عنوان: ASP.NET MVC #1
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۰۳ ۰۸:۳۲:۰۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

چرا ASP.NET MVC ؟

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 MVC نسبت به ASP.NET web forms

1) سادگی نوشتن آزمون‌های واحد

مهم‌ترین دلیل استفاده از ASP.NET MVC صرفنظر از تمام دلایل دیگر، بحث طراحی ویژه آن جهت ساده سازی تهیه آزمون‌های واحد است. مشکل اصلی نوشتن 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 وهله 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 نوشتن آزمون واحد برای آن غیرممکن نباشد، به همین دلایل آنچنان مرسوم هم نیست.

البته شاید بپرسید که این مساله چه اهمیتی دارد؟ امکان نوشتن ساده‌تر آزمون‌های واحد مساوی است با امکان ساده‌تر اعمال تغییرات به یک پروژه بزرگ. تغییرات در پروژه‌های بزرگی که آزمون واحد ندارند واقعا مشکل است. یک قسمت را تغییر می‌دهید، 10 قسمت دیگر به هم می‌ریزند. اینجا است که مدام باید به کارفرما گفت: «نه!»، «نمیشه!» 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!»

2) دستیابی به کنترل بیشتر بر روی اجزای فریم ورک

در طراحی ASP.NET MVC همه‌جا interface ها قابل مشاهده هستند. همین مساله به معنای افزونه پذیری اکثر قطعات تشکیل دهنده ASP.NET MVC است؛ برخلاف ASP.NET web forms. برای مثال تابحال چندین routing engine، view 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 آن‌را دوست ندارید؟ آن‌را با یک نمونه بهتر تعویض کنید و الی 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 نوشتن آزمون‌های واحد.

3) سرعت بیشتر اجرا

ASP.NET MVC یک سری از قابلیت‌های ذاتی ASP.NET web forms را مانند ViewState 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 دسکتاپ 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 کیلوبایت یا بیشتر هم برسد. همین مساله بر روی سرعت دریافت و اجرا تاثیر گذار خواهد بود.

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

خوب، ViewState حذف شده، بنابراین اکثر کنترل‌های ASP.NET web forms هم کاربرد آنچنانی در ASP.NET MVC نخواهند 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 کاملا از نو بازنویسی شده خواهید 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.

5) 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 منحصربفردی تولید می‌کرد که جهت کار با فریم ورک‌های جاوا اسکریپتی عموما 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 سنخیتی با ViewState و Postback وب فرم‌ها ندارد و همین مساله عموما مشکل‌زا است. علاوه بر آن اخیرا مایکروسافت توسعه ASP.NET Ajax خود را تقریبا در حالت تعلیق و واگذار شده به شرکت‌های ثالث درآورده است و توصیه آن‌ها استفاده از jQuery Ajax است. اینجا است که مدل ASP.NET 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 ایی است که این روزها همه‌جا علیه آن صحبت می‌شود. البته یک سری آداپتور 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 حذف شده، استفاده از این افزونه‌ها مشکل ساز نخواهد بود.

6) استفاده از امکانات جدید زبان‌های دات نتی

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 datatable و dataset بوده است (با توجه به امکانات زبان‌های دات نتی در آن دوران). بنابراین اگر علاقمند هستید که این امکانات جدید را بکاربگیرید، ASP.NET MVC برای استفاده از آن‌ها طراحی شده است!

7) 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 کنترلرها 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 الی 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 بکارگرفته 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 ASP.NET MVC شروع کنید.

8) 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 مسایل، 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 آنچنان تعداد بالایی کنترل و افزونه از پیش مهیا شده نیست. در بدو امر شما فقط یک سری html helper، url helper و ajax helper ساده را خواهید دید؛ این نقطه ضعف ASP.NET MVC نیست. عمدا 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 باشید تا آنچنان 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 الی 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 منتظر 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 زیرساخت 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 ۱۱:۵۹:۱۸

واقعا خیلی خوشحالم که مباحث mvc رو شروع کردید. من برای اولین بار سعی کردم یه مبحث رو به طور اصولی پیگیری کنم و سئوالات ابهاماتی برام پیش اومده که مطمئنم اینجا 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 !
ممنون.

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

sسلام 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 خیلیا 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 را تازه شروع کنیم از MVC 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 انگیزه من رو برای یادگیری بیشتر کرد، فکر کنم هرکس 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 رسمی asp بخش جالب http://www.asp.net/mvc/books رو داره که من کتاب Professional ASP.NET MVC 3 (Wrox Programmer to Programmer) رو انتخاب کردم و ازشم راضیم .

در مورد مدارک هم می دونم که فعلا فقط توی امتحان Exam 70-515 برای مدرک TS: Web Applications Development with Microsoft .NET Framework 4 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 قمشلویی
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 آرزوم 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 میشه 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 Vlidation در MVC شد :)) جالبه وقتی javascript بر روی مرورگر Disable باشه Validation سمت سرور کار نمیکنه. و آخرشم راه های عجیب و غریب برای رفع مشکل یافتم. اما در web form ها اکثر این مشکلات ابتدایی را قبلاً پشت سر گذاشته ایم. (البته گفتم ماهیت IT یادگیری یادگیری و یادگیری است)

یکی دیگر از مشکلات مهاجرت، عدم وجود Design Preview موقع استفاده از Razor است که کمی کار را مشکل میکند و شما یا 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، شایدم من اشتباه میکنم.
نظر شما چیست؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۰۶ ۱۵:۴۱:۳۰

- هر وقت یادگرفتید («همه» منظورم هستند! نه شخص خاصی)، این تجربیات کوچک خودتون رو در قالب یک مطلب وبلاگ 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 پویا سروکار دارید و نه با یک HTML ثابت.

نویسنده: Shima Besharat
تاریخ: ۱۳۹۱/۰۱/۰۶ ۱۵:۵۰:۱۴

ممنون از پاسخگویی و دقت شما.
- در خصوص وبلاگ 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 پاسخ شما کاملاً منطقیه.

پ.ن:
آقا وحید چند روزیه اعصاب نداریاااا :) :) :) :}

نویسنده: Majid Shamkhani
تاریخ: ۱۳۹۱/۰۱/۰۷ ۰۲:۰۱:۰۴

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

نویسنده: Std
تاریخ: ۱۳۹۱/۰۱/۰۷ ۱۳:۲۷:۳۵

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

نویسنده: amir
تاریخ: ۱۳۹۱/۰۵/۱۴ ۱۸:۲۷

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 datareader و dataadapter 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

dataset مربوط به دات نت 2 است. لازم است یک سری پیشنیاز رو اول مطالعه کنید:
وضعیت فناوری‌های مرتبط با دات نت از دیدگاه مرگ و زندگی!
5 دلیل برای استفاده از یک ابزار ORM
امنیت در LINQ to SQL (برای مابقی ORMها هم به همین صورت است)
مروری بر کدهای کلاس SqlHelper

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 مثال‌های MVC این سری:
MVC_Samples

نویسنده: محمدرضا
تاریخ: ۱۳۹۱/۱۰/۰۸ ۹:۵۹

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
ممنون از سایت خوبتان :)

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

آیا به غیر از در نظر گرفتن حذف ViewState در برنامه سرعت نمایش صفحات در Asp.net mvc بیشتر است یا Asp.net Web 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 نمیشن ؟ یا هر کاربر در ورود اولیه خود به سایت با این تاخیرها در نمایش مواجه است ؟

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

- در نظرات مطلب « PHP سریعتر از ASP.NET! افسانه یا واقعیت؟ » در مورد علت تاخیر اولیه نمایش سایت‌های ASP.NET بحث شده . (چند مرحله‌ای بودن کامپایل کدهای دات نت و زمان بارگذاری اولیه برنامه در یک پروسه وب؛ بعد از این گذر اولیه الزامی،

با سرعت خوبی مواجه خواهید بود)
- ضمنا 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: ۱۳۹۲/۰۸/۰۹ ۱۶:۳۹

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

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

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 فاطمی
تاریخ: ۱۳۹۲/۰۹/۱۴ ۱۶:۲۵

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 ندیدم و شما باید با پیکسل‌ها کار کنید . آیا این درسته ؟

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

- ASP فرق می‌کنه با ASP.NET؛ ASP 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 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 برده 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. مثلا twitter bootstrap 3 یک فریم ورک CSS اصطلاحا 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؟
ASP.NET یک فناوری سمت سرور است که نهایتا 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 اساسا این مباحث (مثلا طراحی واکنش‌گرا) ربطی به فناوری‌های سمت سرور ندارند و جزو مباحث سمت کاربر محسوب می‌شوند.

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

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

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

مسیر راه « ASP.NET MVC »

نویسنده: پاییز
تاریخ: ۱۳۹۳/۱۲/۱۴ ۱۹:۳

در پاراگراف آخر صحبتی از asp.net واحد شده بود و اینکه سمت و سوی آینده مایکروسافت چیزی است به اسم web API.

سوال من این است که :
-الان وضعیت این Web API به کجا رسیده است؟ با همان دید اولیه پیش رفته است؟
-آیا اینکه مثلا اگر اکنون Web API به بلوغ رسیده است، پس شروع کنیم به جای یادگیریasp.net mvc به سمت یادگیریweb API برویم، درست است؟ یا کلا یادگیری اینها دو مقوله جدا از هم است؟

متشکرم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۲/۱۴ ۱۹:۱۱

- مطالب « گروه ASP.NET Web API » را در سایت پیگیری کنید.
- دو گروه مجزا هستند. ASP.NET Web API جایگزین مدرنی است برای WCF Rest Services. در ASP.NET MVC 6 این دو با هم یکی خواهند شد و در یک کنترلر امکان استفاده از امکانات متداول ASP.NET MVC به همراه متدهای Web API میسر خواهد شد.

نویسنده: پاییز
تاریخ: ۱۳۹۳/۱۲/۱۴ ۱۹:۱۸

در ورژن‌های جاری ASP.NET MVC امکان استفاده از متدهای Web API در کنترل‌های ASP.NET MVC فراهم نیست؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۲/۱۴ ۱۹:۲۵

ASP.NET Web API وابستگی به ASP.NET MVC ندارد و در برنامه‌های وب فرم هم قابل استفاده‌است (یا حتی در یک سرویس ویندوز NT). در برنامه‌های ASP.NET MVC فعلی امکان استفاده‌ی از امکانات آن در Viewهای برنامه وجود دارد. این جزئیات در مطالب « گروه ASP.NET Web API » بیشتر بررسی شده‌اند.

عنوان: ASP.NET MVC #2
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۰۴ ۱۷:۴۴:۰۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

MVC چیست و اساس کار آن چگونه است؟

الگوی MVC در سال‌های اول دهه 70 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 2004 برای اولین بار در سکویی به نام Rails به کمک زبان روبی جهت ساخت یک فریم ورک وب MVC مورد استفاده قرار گرفت و پس از آن به سایر سکوها مانند جاوا، دات نت (در سال 2007)، 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 MVC 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 وهله سازی Model را عهده دار خواهد شد و حاوی اطلاعاتی است که نهایتا 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 ارکستر الگوی 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 خطایی 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، صف‌ها 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 MVC با ASP.NET Web forms

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 وهله سازی شده معرف آن صفحه را ایجاد کرده و آن‌را اجرا می‌کند. در اینجا چرخه طول عمر صفحه مانند page_load و غیره رخ خواهد داد. جهت انجام این وهله سازی، View به فایل code behind خود گره خورده است و جدا سازی خاصی بین این دو وجود ندارد. منطق صفحه به markup آن که معادل است با یک فایل فیزیکی بر روی سیستم، کاملا 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 ایی در سیستم. سپس این کنترلر کار پردازش درخواست رسیده را شروع کرده، اطلاعات مورد نیاز را جمع آوری و سپس به View ایی که انتخاب می‌کند، جهت نمایش نهایی ارسال خواهد کرد. در اینجا View این اطلاعات را دریافت کرده و نهایتا 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 آن‌را 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 آن‌را به صورت خودکار وهله سازی کرد).

3- این کلاس باید اینترفیس استاندارد IController را پیاده سازی کرده باشد.

4- و نام آن باید مختوم 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 تصور 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: ۰۱:۱۲:۲۰ ۱۳۹۱/۰۱/۰۵

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: ۰۸:۵۱:۳۶ ۱۳۹۱/۰۱/۰۵

بله. در کل هیچ قید خاصی نداره این قسمت.

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

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: ۱۷:۳۹:۲۳ ۱۳۹۱/۰۲/۰۱

instance = وهله
instantiation = وهله سازی

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

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

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

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

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

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

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

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

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

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.

عنوان: ASP.NET MVC #3
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۰۶ ۱۶:۳۰:۰۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

تهیه پیش‌نیازهای شروع به کار با ASP.NET MVC

در زمان نگارش این مطلب، نگارش نهایی ASP.NET MVC 3 در دسترس است و همچنین نگارش بتای 4 آن نیز قابل دریافت و نصب می‌باشد. بنابراین فعلا اساس را بر مبنای نگارشی قرار خواهیم داد که در محیط کاری قابل استفاده باشد.
ASP.NET MVC 3 پس از ارائه Visual Studio 2010، منتشر شد و VS.NET به صورت پیش فرض به همراه ASP.NET MVC 2 است.
ساده‌ترین روش نصب ASP.NET MVC 3 بر روی VS 2010 استفاده از برنامه رایگانی است به نام Web Platform Installer. این برنامه را از این آدرس می‌توان دریافت کرد: http://microsoft.com/web/downloads
پس از دریافت آن حداقل دو راه برای نصب ASP.NET MVC 3 وجود دارد. یا گزینه‌ی نصب ASP.NET MVC 3 Tools Update را انتخاب کنید و یا سرویس پک یک VS 2010 را از طریق این برنامه یا جداگانه (بسته کامل و مستقل) دریافت و نصب نمائید. VS 2010 SP1 نیز به همراه ASP.NET MVC 3 است؛ همچنین IIS Express را که نسخه ساده شده IIS 7.5 مخصوص توسعه دهنده‌ها است، می‌توان با این نگارش یکپارچه کرد.

بنابراین به صورت خلاصه بهترین کار این است که سرویس پک یک VS 2010 را یکبار نصب 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 آن‌را دریافت و نصب نمائید (^). البته نصب IIS Express در اینجا یک گزینه اختیاری است و الزامی نیست.

مروری بر ساختار یک پروژه ASP.NET MVC

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

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

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

فعلا جهت توضیحات اولیه بیشتر، گزینه‌ی Internet Application و نوع View Engine را هم 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 تفاوت دارند (به آن templating language هم گفته می‌شود). نوع ASPX همان Syntax متداول قدیمی ASP.NET را تداعی می‌کند و نوع Razor به صورت اختصاصی برای ASP.NET MVC تهیه شده است.

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

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

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

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

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

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

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

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

قرار دادهای پوشه‌های پیش فرض یک پروژه ASP.NET MVC

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

نظرات خوانندگان

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

مرسی از مطلب کاملتون. امیدوارم ادامه بدین سریعتر.
سینتکس Razor اختصاصا برای ASP.NET Web Pages در IDE اون یعنی WebMatrix توسعه داده شده بود.اما بعدا به‌خاطر سهولت در استفاده و استخون‌بندی web pages ها در مقایسه با فریم‌ورک MVC توی نسخه 3 ام‌وی‌سی استفاده شد.

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

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

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

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

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

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

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

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

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

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

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

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

کلاس ProductController در پوشه Controllers اضافه 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: ۲۳:۵۵:۰۸ ۱۳۹۱/۰۱/۰۷

ممنون
اما یه مشکلی که هست اینه که من به طور مثال یه کنترل به نام SendController.cs ایجاد کردم با یک ViewResult به نام index.
حالا زمانی که برای این متد index یک view میسازم، خودش به صورت خودکار یک پوشه با نام send در پوشه ی View ایجاد میکنه.
View/send/file ha
منتها من میخوام که این پوشه ی send داخل پوشه ی دیگه ای باشه مثل : Administrator که خودم ایجادش کردم.
View/Administrator/send/file ha
در صورت تعویض جاها به صورت دستی قاطی میکنه :)

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

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

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

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

ASP.NET MVC 3 RTM

ASP.NET MVC 3 Tools Update

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

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

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

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

نویسنده: vahid21
تاریخ: ۱۷:۶ ۱۳۹۱/۰۴/۱۶

درود

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

نویسنده: وحید نصیری
تاریخ: ۱۷:۹ ۱۳۹۱/۰۴/۱۶

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

نویسنده: ندا صابری

تاریخ: ۱۳۹۳/۰۱/۰۱ ۹:۴۳

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: ۱۳۹۳/۰۱/۰۱ ۱۰:۵۹

- چون مطابق آمار کسی با موتور وب فرم‌ها در MVC پس از نگارش 3 کار نمی‌کند، گزینه‌ی آن‌را 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 .

عنوان: ASP.NET MVC #4
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۰۸ ۰۰:۴۷:۰۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

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 انتخاب کرده (ASP.NET MVC 3 Web Application و بعد انتخاب قالب Empty نمایش داده شده) و سپس پروژه جدیدی را شروع می‌کنیم. View Engine را هم Razor در نظر خواهیم گرفت.
پس از ایجاد ساختار اولیه پروژه، بدون اعمال هیچ تغییری، برنامه را اجرا کنید. بلافاصله 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 مراجعه 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. جریان چیست؟!

سلسله مراتبی که بکارگرفته شده به صورت زیر است:

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

ControllerBase، اینترفیس IController را پیاده سازی کرده و سپس کلاس Controller از کلاس ControllerBase مشتق شده 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 متدی یا خاصیتی در نگارش بعدی به این اینترفیس اضافه شد، هیچکدام از پروژه‌های قدیمی دیگر کامپایل نخواهند شد و باید ابتدا این متد یا 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 تعریفی‌ام را پیاده سازی کنید! همینه که هست! اما اگر طراح مایکروسافت باشد، بلافاصله 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 متدی در اینترفیس IController بنابر نیازهای جدید درنظر گرفته شد، بتوان سریع یک پیاده سازی پیش فرض از آن‌را 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

Home، به دنبال یک سری فایل می‌گردد. فایل‌های aspx مربوط به View Engine ایی به 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 ایی 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 گرنه vbhtml ایجاد می‌شد):

```
@{
    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 انتخاب شد. چرا مثلا نام TestController وارد نشد؟ برنامه از کجا متوجه شد که باید حتما این کنترلر را پردازش کند. نقش متد Index چیست؟ آیا حتما

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 متد 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
defaults
            );
}
```

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 ASP.NET MVC از این امکانات برای ارسال درخواست‌ها به کنترلرها استفاده می‌شود.

در متد routes.MapRoute فراخوانی شده‌ای که در کدهای بالا ملاحظه می‌کنید، کار نگاشت یک URL به Actionهای 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 متد 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 آن‌را به عنوان نام کنترلر تفسیر خواهد کرد، دومین قسمت آن، نام متد عمومی موجود در کنترلر فرض شده و سومین قسمت به عنوان پارامتر ارسالی به این متد پردازش می‌شود.
برای مثال از آدرس زیر اینطور می‌توان دریافت که:

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

Home نام کنترلی است که فریم ورک به دنبال آن خواهد گشت تا این درخواست رسیده را پردازش کند و about نام متدی عمومی در این کلاس است که به صورت خودکار فراخوانی می‌گردد. در اینجا پارامتر id ایی هم وجود ندارد. در یک برنامه امکان تعریف چندین و چند مسیریابی وجود دارد.
پارامتر سوم متد 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 http://localhost/، هر سه مقدار پیش فرض مورد استفاده قرار خواهند گرفت چون سه جزئی ({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 متد routes.MapRoute خواهد بود و متد پیش فرض نیز Index و پارامتری هم به آن ارسال نخواهد شد.
در حالت http://localhost/home نام کنترلر مشخص است اما دو جزء دیگر ذکر نشده‌اند، بنابراین مقادیر آن‌ها از پیش فرض‌های صریح ذکر شده در متد routes.MapRoute گرفته می‌شود. یعنی نام متد 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 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 وهله سازی و سپس متدی به نام List در آن فراخوانی خواهند شد.

یک نکته:
ترتیب routes.MapRouteهای 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. مثلا 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: ۱۳۹۱/۰۱/۰۸ ۱۵:۴۲:۱۷

با سلام و تشکر از مهندس عزیز

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

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۰۸ ۱۶:۵۶:۴۲

از اینجا می‌تونید دریافت کنید: (^)
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: (^)

نویسنده: Amin Noorivandi
تاریخ: ۱۳۹۱/۰۱/۰۸ ۲۱:۵۶:۲۳

وحید جون داداش سرعتت خیلی کمه. ما الان تو ویو هستیم. توروخدا پرسون جزوه بعدیتو

نویسنده: Shima Besharat
تاریخ: ۱۳۹۱/۰۱/۰۹ ۰۰:۱۰:۲۲

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

نویسنده: Hossein Raziee
تاریخ: ۱۳۹۱/۰۱/۰۹ ۰۰:۵۳:۰۵

اتفاقا جناب نصیری خیلی سریع هم مینویسند.
واقعا دستشون درد نکنه.

نویسنده: Arezoo A
تاریخ: ۱۳۹۱/۰۱/۱۲ ۲۱:۵۰:۳۹

یک دنیا تشکر.مثل همیشه عالیه

نویسنده: TazeKar
تاریخ: ۱۳۹۱/۰۳/۳۱ ۱۲:۰

با سلام و تشکر فروان
من هم مانند دوستان خیلی علاقمندم که از setting استفاده کنم که متاسفانه لینک دانلود خراب است لطفا 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 رو انتخاب کنید. به این صورت درست دریافت خواهد شد.

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

سلام.با تشکر از زحمات شما.نحوه فعال کردن settingرو برای vsتوضیح میدین.ممنون

نویسنده: 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 برامون راحت‌تر کرده.بازم متشکرم.

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

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

نویسنده: Ara
تاریخ: ۱۳۹۱/۰۴/۰۱ ۱۶:۳۷

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

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

درود 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

نویسنده: 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 بدهم 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: ۱۳۹۱/۰۵/۰۴ ۱۳:۵۷

return View چندین و چند overload دارد. حالت پیش فرض آن استفاده از viewایی هم نام اکشن متد 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

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

مگر اینکه بخواهید redirectهای خاصی را تعریف کنید. مثلا 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 نوشت:

```
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 خواهند 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 اکشن متدها 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 متدی در آن کنترلر (در صورت وجود) و الی آخر. بنابراین نیازی نیست به ازای هر کنترلر جدید یک route نوشت؛ یا به ازای هر متدی در آن به همین ترتیب.

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

خیلی ممنون

نویسنده: 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

```
http://site.ir/مرتبط لینک‌های
```

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

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

- در انتهای قسمت 9 در مورد لینک‌های فارسی توضیح دادم: (^)
- همچنین شما در MVC می‌تونید 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

نویسنده: اوبالیت
تاریخ: ۱۳۹۱/۱۱/۰۴ ۱۳:۲

درود؛ دو سوال داشتم.
1) هنگامی که من می‌خواستم یک View رو 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؟

نویسنده: 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. ضمنا ViewBag یک شیء Dynamic است و کلا برای آن intellisense وجود ندارد.

نویسنده: 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؟

-2اگر قرارداد درست بود و ما در قسمت دوم مسیر پیش فرض یکی از بخش‌ها رو مشخص نکرده باشیم چه اتفاقی میافته؟
مثلاً 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. متد دیگری انتخاب نمی‌شود. اولویتی برای انتخاب اکشن متد بعدی وجود ندارد؛ بجز مواردی که صریحا 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 متدی دارم که ورودی آن int و خروجی string ای را که از دیتابس میگیرد را بر میگرداند. مشکلم اینه که ورودی متد رو باید از view و بر اساس مقداری که کاربر در حین اجرای برنامه تو textbox وارد میکنه بگیرم و خروجی متد را در view به کاربر نشون بدم. چیکار باید بکنم؟ البته مقدار عددی یکی از propertyهای 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 پیشنیاز لازم داشته). روش دیگر انجام این‌کارها استفاده از Ajax است که در قسمت 21 به آن پرداخته شده.

نویسنده: هومن
تاریخ: ۱۳۹۳/۰۷/۲۸ ۱۷:۲۷

با سلام و خسته نباشید و تشکر بابت مقاله مفیدتون ، بنده سوالاتی داشتم
1 - چطور میتونم در برنامم بعد از کامپایل(در زمان اجرا) روت جدید تعریف کنم ، یا به عبارتی چطور میشود مسیری را بصورت داینامیک تعریف کرد؟
2 - فرض کنید یک پلتفرم(نمیدونم اسمشو چی بزارم) داریم + چندین ماژول ، ماژول‌ها application هایی مجزا باشند که بصورت جدا گانه ساخته شده اند و بعد برای پلتفرم اصلی نصب شده اند. پلتفرم اولیه چطور باید درخواست هایی را مدیریت کند که برای ماژول هایی که بعدا طراحی میشوند هستند؟
مدت هاست 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: ۱۳۹۳/۰۷/۲۸ ۱۹:۱۷

ممنون برای پاسختون
اگر امکانش هست درمورد سوال دومم بیشتر توضیح دهید .
اینکه برنامه ری استارت شود مساله ای 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: ۱۳۹۳/۰۷/۲۸ ۲۰:۴

RouteTable.Routes یک شیء با طول عمر singleton است (پارامتر ورودی متد RegisterRoutes در مطلب فوق که در فایل global.asax.cs مقدار دهی می‌شود). یک وهله از آن در طول عمر برنامه با تمام اجزای آن وجود دارد. هر جزئی که اطلاعاتی را به RouteTable.Routes 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.

عنوان: ASP.NET MVC #5
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۰۹ ۰۹:۵۳:۰۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

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

در ASP.NET Web forms در فایل code behind یک فرم مثلا 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 ایی که به آن اضافه کرده‌ایم. همچنین مطابق روشی که ذکر شد، View ایی 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

ViewBag 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. مثلا 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 جاری 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 وهله از این شیء را در متد Index ایجاد کرده و سپس به view متناظر با آن ارسال می‌کنیم (در قسمت return View کد زیر مشخص است). بدیهی است این وهله سازی در عمل می‌تواند از طریق دسترسی به یک بانک اطلاعاتی یا یک وب سرویس و غیره 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) متد کمکی 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 متد، 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 آن‌را 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، Intellisense ویژوال استودیو ظاهر می‌شود. به این معنا که شیء Model بکارگرفته شده اینبار دیگر dynamic نیست و دقیقا 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 دقیقا 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 نکرده‌اید تا VS.NET بتواند با اعمال Reflection بر روی اسمبلی برنامه آن‌را پیدا کند. سپس در قسمت Scaffold template گزینه List را انتخاب کنید تا Code generator توکار VS.NET فعال شود. اکنون بر روی دکمه Add کلیک نمائید 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 ViewData 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 متد 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.

احتمالا 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

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 View ها نیز در دسترس خواهد بود. البته TempData یک سری تفاوت هم با سشن معمولی ASP.NET دارد:
- بلافاصله پس از خوانده شدن، حذف خواهد شد.
- پس از پایان درخواست از بین خواهد رفت.
هر دو مورد هم به جهت بالابردن کارآیی برنامه‌های ASP.NET MVC و مصرف کمتر حافظه سرور درنظر گرفته شده‌اند.
البته کسانی که برای بار اول هست با ASP.NET مواجه می‌شوند، شاید سؤال بپرسند 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 دسکتاپ 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 دقیقا مشکل تمام تازه واردها به دنیای وب است که چرا اشیاء ما نیست و نابود شدند. در اینجا وب سرور قرار است به هزاران درخواست رسیده 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 متد جاری 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 خطاها 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 متد جدید به نام‌های InsertData و ShowError اضافه شده‌اند. در متد InsertData ابتدا بررسی می‌شود که آیا نامی وارد شده است یا خیر. اگر خیر توسط متد RedirectToAction، کاربر به اکشن یا متد 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 گزینه Add view برای اضافه کردن View مرتبط با آن اکشن).
محتوای View مرتبط با متد افزودن اطلاعات فعلا مهم نیست، ولی View نمایش خطاها در ساده‌ترین حالت مثلا 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 تکمیلی در مورد Strongly typed viewها:

عنوان شد که Strongly typed view روش مرجح بوده و بهتر است از آن استفاده شود، زیرا اطلاعات اشیاء و خواص تعریف شده در یک View تحت نظر کامپایلر قرار می‌گیرند که بسیار عالی است. یعنی اگر در View بنویسم FirstName: @Model.FirstName1 چون FirstName1 وجود خارجی ندارد، برنامه نباید کامپایل شود. یکبار این را بررسی کنید. برنامه بدون مشکل کامپایل می‌شود! اما تنها در زمان اجرا است که صفحه زرد رنگ معروف خطاهای ASP.NET ظاهر می‌شود که چنین خاصیتی وجود ندارد (این حالت پیش فرض است؛ یعنی کامپایل یک View در زمان اجرا). البته این باز هم خیلی بهتر است از ViewBag، چون اگر مثلا ViewBag.Country1 را وارد کنیم، در زمان اجرا تنها چیزی نمایش داده نخواهد شد؛ اما با روش Strongly typed view، حتما خطای Compilation Error به همراه نمایش محل مشکل نهایی، در صفحه ظاهر خواهد شد.
سؤال: آیا می‌شود پیش از اجرای برنامه هم این بررسی را انجام داد؟
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 متنی ساده مثلا 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. مجددا بر روی این پروژه Unload شده کلیک راست نموده و گزینه edit را انتخاب نمائید. در صفحه باز شده، MvcBuildViews را یافته و آن‌را true کنید. سپس پروژه را Reload 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 ASP.NET MVC از MS-PL 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 متداول یک پروژه سورس باز که شامل دریافت تغییرات و وصله‌ها از جامعه برنامه نویس‌ها است، تغییر کرده است (^ و ^).

نظرات خوانندگان

نویسنده: علی قمشلویی
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 متد ShowError وجود داره که شما قبل از استفاده برای NULL یا خالی بودن چکش کردید؟

ضمنا درخواستی داشتم من تو وب تازه کارم برا
همین CSS وHTML رو خوب بلد نیستم دنبال منبع آموزشی برای
یادگیریشون هستم.البته میخوام مواردی رو که بیشتر درMVC 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: (A Better CSS: LESS and 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 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 کنترلرهای 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 شه خاصیت casecade رو مطابق نیاز تغییر داد.

مثلا وقتی یکی از RelatedCustomers ها رو clear می کنم بدلیل خاصیت "cascade="all,delete-orphan علاوه بر ارجاعات خود Customer هم در هنگام ذخیره سازی پاک میشه.

ممنون میشم راهنماییم کنید.

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

سلام، این مورد در دو کلاس ManyToManyConventions و NamingConventions قابل تنظیم است. پیش فرض آن‌ها بر این مبنا 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 را جستجو کنید. یک enum ساده است. این رو تغییر بدید به حالتی که مدنظر است. شبیه به همین مورد در کلاس NamingConventions در دو متد ReferenceConvention و OneToManyConvention وجود دارد.

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

تبریک 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: ۱۳۹۱/۰۱/۱۰ ۱۲:۴۵:۲۸

من نگفتم MS-PL متن باز نیست. این مجوز پذیرفته شده OSI است (^) .
متن باز بودن هم به معنای آزادی مطلق نیست. مثلا مجوز GPL به شما می‌گه 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 می‌گه من نمی‌خوام و مهم نیست؛ یک تشکر برای من کافی است.
مجوز MS-PL بیشتر معنای (Shared source) رو از طرف مایکروسافت داره. به عبارتی مجاز هستید در فرم باینری 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 باینری 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 کردند و از حالت shared code فقط خواندنی بیشتر جهت بررسی‌های امنیتی و مرور کد، به یک حالت پویاتر تبدیل شده.

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

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 کنترلرها و Viewها نیز در دسترس خواهد بود " .پس علت استفاده از کد زیر برای چیست؟

```
ViewBag.Erorr=erorr;
```

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 وهله‌ای 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 Html.Raw و MvcHtmlString.Create توضیح بدید .

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

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

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

نویسنده: امین
تاریخ: ۱۳۹۱/۰۷/۲۲ ۱۸:۱۸

فرق

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

با

```
private string firstName;

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

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 کنترلرمون مقادیریمون را در قالب alert تو view 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؟
اصلا 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 می‌تونید 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 ;ViewData["DateTime"] = "<br/>" + DateTime.Now

اشاره کردید و راهکار نمایش آنهم را نشان دادید .Html.Raw(myString@)

حال اگر در view در بخش foreach به جای این @Html.DisplayFor(modelItem => item.FirstName از Html.Raw 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 (^)WebViewPage. این کلاس حاوی تعاریف اولیه TempData، ViewBag، ViewData و ... Model است؛ جهت استفاده در Viewها. در کنترلرها هم این تعریف در کلاس پایه ControllerBase قرار دارد.

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

در مورد فعال کردن MvcBuildViews من وقتی این کارو انجام میدم با خطای زیر روبه رو 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_krwrdmvd.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
```

نویسنده: طاهری
تاریخ: ۱۳۹۲/۰۹/۰۶ ۱۰:۲۵

سلام
Employees رو ایجاد کردم.متد List روهم در کنترلر گذاشتم.ویو رو همونطور که گفتین add کردم(دقیقا 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*

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

اشاره‌گر رو داخل اکشن متد کنترلر قرار بدید و بعد پروژه رو در دیباگر VS.NET اجرا کنید. مراحل کاری ASP.NET MVC از View شروع نمی‌شود. مراجعه کنید به «تفاوت مهم پردازشی ASP.NET MVC با ASP.NET Web forms» در قسمت دوم این سری.

نویسنده: آروین
تاریخ: ۱۳۹۲/۰۹/۲۱ ۱۲:۵۳

با عرض سلام و خسته نباشید
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 هرکدام از اینها مقادیر را از دیتابیس بگیرم. در روش 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 ViewModel مخصوص همان PartialView هستند ولی ارتباط با دیتابیسم بیشتر میشه. ممنون می‌شم یه توضیحی در مورد سرعت و کارایی هرکدام از روش‌ها بدین و آیا در یک پروژه وب سایت روش اول به صرفه‌تر است یا دوم؟
با تشکر

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

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

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

سلام. ببخشید بعداز اجرای view index, میخواهم view list رو اجرا کنم. view list اجرا نمیشه. با هر بار اجرا همون view index اجرا میشه. باتشکر.

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

- 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 دیباگ 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 داره ؟

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

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

مثلا برای حالت anonymously typed در پارامتر route values آن (این پارامتر مدل نیست؛ مقادیر route 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 برام لاینحل مونده ممنون میشم راهنماییم کنین اونم عدم وجود Desing mode برای مثلا مستر پیج یا صفحات دیگرمون هست یعنی الان کاملا ذهنی صفحات رو طراحی میکنم و از اونجایی که موتور Razor خیلی ساده‌تر از Aspx هست نمیخوام بذارمش کنار. برای این مساله یه راه ساده‌تری وجود نداره که دیزاین طراحی پیجامون رو هم داشته باشیم؟

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

- Page Inspector برای اینکار طراحی شده.
- 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 عموما 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 کنید (نیازی به کامپایل مجدد نیست).
- صفحات ASP.NET، یک سری صفحات پویا هستند. نیاز به برقراری اتصالات خاصی بین بانک اطلاعاتی، کوئری استرینگ‌ها، مقادیر 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.

نویسنده: 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

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

در سی‌شارپ فعلی برای مقدار دهی به خواص خودکار و تعیین مقدار پیش فرض، از مقدار دهی آن‌ها در سازنده کلاس استفاده می‌کنند (یک constructor تعریف کنید و بعد خواص را مقدار دهی کنید). این مورد در سی‌شارپ 6 قرار است ساده‌تر شده و در همان محل تعریف خاصیت، قابل مقدار دهی شود.

عنوان: ASP.NET MVC #6
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۱۱ ۰۰:۰۸:۰۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

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 باشید که در یک کلاس کنترلر، متدی عمومی را از این سیستم خارج کنید، تنها کافی است آن‌را با ویژگی (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 متد 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 متد 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.

انواع Action Results در ASP.NET MVC

در ASP.NET MVC بجای استفاده مستقیم از شیء Response، از شیء ActionResult جهت ارائه خروجی یک متد استفاده می‌شود و مهم‌ترین دلیل آن هم مشکل بودن نوشتن آزمون‌های واحد برای شیء Response است که وهله 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).
سلسه مراتب ActionResultهای قابل استفاده در ASP.NET در تصویر زیر مشخص شده‌اند:

و در مثال زیر تقریبا انواع و اقسام ActionResultهای 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) مثلا متد GetJavaScript را درنظر 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 سلسله مراتبی ActionResultها می‌توان جستجو کرد. تمام این کلاس‌ها نوعی ActionResult هستند و از یک کلاس پایه به ارث رسیده‌اند.

2) مثلا 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 اکشن متدها 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. مثلا 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 پیش فرض درون یک ContentResult محصور خواهد شد.

5) اگر خروجی متدی در اینجا از نوع void باشد، با ActionResult ایی به نام EmptyResult یکسان خواهد بود. بنابراین با متدهای SendMail و SendMailCompleted به یک نحو رفتار می‌گردد.

6) return Json یاد شده که خروجی‌اش از نوع JsonResultاست در پیاده سازی‌های Ajax ایی کاربرد دارد.

7) جهت بازگرداندن حالت وضعیت 403 یا غیرمجاز می‌توان از return new HttpUnauthorizedResult استفاده کرد.

8) یا جهت اعلام مشکلی در سمت سرور به کمک return new HttpStatusCodeResultکد ویژه‌ای را می‌توان به کاربر نمایش داد.

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

و خلاصه اینکه تمام کارهایی را که پیشتر در ASP.NET Web forms ، مستقیما به کمک شیء Response انجام می‌دادید (Response.Redirect، Response.End، Response.Write و غیره)، اینبار به کمک یکی از ActionResultهای 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 قمشلویی
تاریخ: ۱۳۹۱/۰۱/۱۱ ۱۸:۳۶:۱۹

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

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۱۱ ۱۹:۰۹:۱۸

یک سری از اشیاء، اشیاء توکار ASP.NET هستند مانند Application، Server، Response، Request و Session و در فضای نام System.Web تعریف شده‌اند. این اشیاء جزو ASP.Net Runtime هستند و در تمام فریم ورک‌هایی که بر این پایه تهیه شده‌اند قابل دسترسی هستند.
برای نمونه کار شیء Response نمایش اطلاعات به کاربر، تغییرات اعمالی بر روی هدر ارسالی (مثلا 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 غیراینصورت نال خواهند بود. برای مثال اگر متد Response.Write در حین یک Unit test فراخوانی شود، قابل آزمایش نخواهد بود مگر اینکه مراحل ست آپ وب سرور و وهله سازی HttpContext طی شود (که کار پر دردسری است). MVC یک لایه abstraction بر روی این اشیاء ایجاد کرده تا در حین انجام آزمون‌های واحد درگیر این مراحل نشویم.

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

با درود، ممنون بابت نظم و پیگیری و دقت نظر در ارائه مطالب مخصوصا این سری ام‌وی‌سی که امیدوارم سریعتر ادامه بدین.
یک سوال: در قسمت دوم مقاله‌های MVCتون ظاهرا بود که گفتین، هدف این فریم‌ورک MVC معطوف به UI/UX هست. ولاغیر.
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 سوالم، اخیرا خیلی میبینم کتابخانه‌هایی با پسوند .js ظهور کردن. آیا اینها همان هدف MVC را نشانه گرفته‌اند و یا اگر نه، چقدر مرتبط و کمک‌کننده هستند؟.
با تشکر مجدد.

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

- به قسمت کار با Ajax در ASP.NET MVC که برسیم، 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: ۱۳۹۱/۰۱/۱۲ ۰۰:۵۱:۴۶

با سلام و تشکر از مطالب محشرتان
ممکنه یک سر نخ در مورد UI/UX بدهید؟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. نمیاد 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. ببینید توی web form طراح میتونه بیاد UI رو بوسیله کنترل ها 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 بده. البته در این مرحله چون توی کد برنامه نیاز به ارتباط با عناصر صفحه وجود داره وابستگی پیش میاد.
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 واقعا جالبه و کار باهاش راحت) ، طراح حتی باید مدل ها رو بشناسه، یا حتی در روش باز هم توصیه شده ، کلاس های سی شارپی به عنوان viewModel تعریف کنه و view رو با اونها تشکیل بده. در حالیکه در مرحله کدنویسی دیگه به عناصر صفحه وابستگی وجود نداره.
منظورم اینه که توی هرکدوم 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 مثلا ستون‌های یک GridView رو طراحی و بایند کنه؟ برای نمونه یک گزارش رو دربیاره؟ اینجا هم به همین ترتیب. یک ترکیب کلی صفحه وجود دارد که طراح می‌تونه صرفنظر از اینکه کار شما PHP است یا ASP.NET در بیاره. مثلا 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 قسمت ها جدا نمیشه.
البته دیدگاه دیگه ای هم وجود داره: این جداسازی تا جایی مهم و موردنظره که نوشتن آزمون های واحد رو راحت تر کنه.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۱۲ ۱۰:۱۹:۵۷

ببینید 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 رو به همراه داره. چون کنترلر ارجاع مستقیمی به اشیاء بصری نداره، برنامه نویس می‌تونه کارش رو بدون نیاز به View پیش ببره و نهایتا 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/

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 به پایان 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 مثلا یک خروجی مشخص به درون مرورگر کاربر Flush شده و درخواست خاتمه می‌یابد. 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 ست آپ سرورمنظور چیست ؟
یعنی MVC پیشفرض نیازی به وب سرور ندارد باتشکر.

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

اگر به TestMethod ایی 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 تشکیل HttpContext ندارد.

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

سلام -
برای آرگومان دوم ("خطا ... "," HttpStatusCodeResult("501" پیام فارسی میذارم بجاش ؟ نشون میده- یونیکد رو پشتیبانی نمی‌کنه یا کار خاصی باید انجام بدم؟
مرسی

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۸/۰۴ ۰:۴۴

مطابق استاندارد ، در HTTP header امکان قرار دادن کاراکترهای یونیکد نیست (پیش فرض آن حداکثر 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 متد 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: ۱۳۹۳/۰۷/۲۰ ۱۳:۱۴

سلام و تشکر؛ میشه لطفا توضیح بدید چرا توی بعضی از بازگشت‌ها مثل EmptyResult و HttpUnauthorizedResult وهله سازی انجام شده (new) و در باقی نیازی به این امر نیست.

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

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

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

در اینجا return new، داخل یک متد کلاس پایه محصور شده‌است.

عنوان: ASP.NET MVC #7
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۱۲ ۱۰:۵۳:۰۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

آشنایی با Razor Views

قبل از اینکه بحث جاری ASP.NET MVC را بتوانیم ادامه دهیم و مثلا مباحث دریافت اطلاعات از کاربر، کار با فرم‌ها و امثال آن‌را بررسی کنیم، نیاز است حداقل به دستور زبان یکی از View Engineهای ASP.NET MVC آشنا باشیم.
MVC3 موتور View جدیدی را به نام Razor معرفی کرده است که به عنوان روش برگزیده ایجاد Viewها در این سیستم به شمار می‌رود و فوق العاده نسبت به ASPX view engine سابق، زیباتر، ساده‌تر و فشرده‌تر طراحی شده است و یکی از اهداف آن تلفیق code و markup می‌باشد. در این حالت دیگر پسوند فایل‌های Viewها همانند سابق ASPX نخواهد بود و به cshtml و یا vbhtml تغییر یافته است. همچنین برخلاف web forms view engine از System.Web.Page مشتق نشده است. و باید دقت داشت که Razor یک زبان برنامه نویسی جدید نیست. در اینجا از مخلوط زبان‌های سی شارپ 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 ، مثلا:

Spark
NHaml
SharpDOM
SharpTiles
Wing Beats
string-template-view-engine-mvc
Bellevue
Brail
Hasic
NDjango

Razor Views 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 viewها به صورت یک کلاس کامپایل می‌شوند و همچنین از System.Web.Page مشتق نخواهند شد، امکان بررسی HTML نهایی تولیدی آن‌هابدون نیاز به راه اندازی یک وب سرور وجود دارد.
3) IntelliSense ویژوال استودیو به خوبی آن‌را 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 HTML آن نیازی به راه اندازی وب سرور و وهله 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 صرفا 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 را به پوشه Controllers 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 را جهت افزودن 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 فعلا از این قسمت صرفنظر نمائید، چون می‌خواهیم آن را دستی ایجاد کرده و توضیحات و نکات بیشتری را بررسی کنیم.

ه) برای اینکه حین اجرای برنامه در VS.NET هربار نخواهیم که آدرس کنترلر Products را دستی در مرورگر وارد کنیم، فایل

Global.asax.cs را گشوده و سپس در متد RegisterRoutes، در سطر Parameter defaults، مقدار پیش فرض کنترلر را مساوی Products 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 />
<img src="@(number).jpg" />
```

در ادامه توضیحات مرتبط با این کدها ارائه خواهد شد:

1) 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 Viewایی است که در قسمت (د) بررسی ساختار پروژه مثال جاری، ایجاد کردیم. در ابتدای آن هم نوع model

مشخص شده تا بتوان ساده‌تر به اطلاعات شیء Model به کمک IntelliSense دسترسی داشت.
برای ایجاد یک قطعه کد در Viewایی 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 نوشتن 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 مثلا 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، دقیقا 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 data تعریف شده در code block قبلی برنامه دریافت و نمایش داده خواهد شد.
شبیه به همین حالت در مثال زیر هم وجود دارد:

```
<img src="@(number).jpg" />
```

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 توئیتری شخصی را وارد کنید. مثلا:

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

در این حالت View کامپایل نخواهد شد و Razor تصور خواهد کرد که قرار است اطلاعات متغیری به نام name را نمایش دهید.
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. حتما 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 {} بدنه 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، احتمالا 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 می‌نوشتیم. مثلا پیشتر یک GridView وجود داشت و یک دیتاسورس که به آن متصل می‌شد؛ مابقی خودکار بود و ما هیچ وقت حلقه‌ای ننوشتیم. در اینجا هم این مساله با نوشتن 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 نوشتن 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 چندجزئی کفایت 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 وهله 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 html نهایی، نیازی به HttpContext نیست. بنابراین از این امکانات مثلا در یک سرویس ویندوز ان تی یا یک برنامه کنسول، WPF، WinForms و غیره هم می‌توان استفاده کرد.
برای اینکه بتوان از Razor خارج از ASP.NET MVC استفاده کرد، نیاز به اندکی کدنویسی هست مثلا استفاده از کامپایلر سی شارپ یا وی بی و کامپایل پویای کد و یک سری ست آپ دیگر. پروژه‌ای به نام RazorEngine این کپسوله سازی رو انجام داده و از اینجا http://razorengine.codeplex.com/ قابل دریافت است.

نظرات خوانندگان

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

لطفا 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 جاهایی 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 مستقیما استفاده کرد.در واقع انعطاف پذیری بیشتری الان وجود داره...

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 (^) WebViewPage. این کلاس حاوی تعاریف اولیه TempData، ViewBag، ViewData و ... Model است. این Model ریشه‌اش به اینجا بر می‌گردد و با حرف بزرگ شروع شده است. بنابراین در Viewهای سی شارپ Razor برای تعریف نوع مدل نیاز است بین model و شیء Model تفاوت وجود داشته باشد.
- در یک View شما هر تعداد مدل رو می‌تونید 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 Strongly typed 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

تاریخ: ۱۳۹۱/۰۱/۱۶ ۲۳:۵۹:۳۱

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 باید 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 واقعا جای توضیحات بیشتر داره که بیصبرانه منتظرش میمونم.
راستش اون موقع که شما آموزش سیلورلایت رو در برنامه داشتید، من اصلا نمی تونستم 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="&amp;" id="&amp;">&</span>
```

چرا مقادیر Attributeها encode شده اما مقدار بین تگ encode 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

- Model ذکر شده در حلقه با M بزرگ 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 خواهند کرد . در mvc با کدام مکانیزم این کار را انجام می‌دهید . من منوه‌های پروژه خودم را با actionlink شبیه سازی کردم . تا اونجایی که من تحقیق کردم در web formsها از مولفه login view 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

else نوشت. مثلا:

```
@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.

نویسنده: میثاق ابراهیمی
تاریخ: ۱۳۹۳/۱۰/۲۲ ۱۴:۶

کدهای view متد index را همانند بالا وارد کردم ولی به تعداد تگ‌های
 به من وارنینگ میدهد:
Unknown element 'br' or element cannot be placed here.
و در مرورگر تگهای br را همانطور نشان میدهد:

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۲۲ ۱۴:۳۴

- دریافت تمام مثال‌های MVC این سری (برای اینکه با copy/paste مثال‌ها مشکلی نداشته باشید): MVC_Samples
+ در متن ذکر شده: «عباراتی که پس از علامت @ معرفی می‌شوند، به صورت پیش فرض Html Encoded هستند (در قسمت 5

در اینباره بیشتر توضیح داده شد)». در قسمت پنجم « یک نکته امنیتی » و متد Html.Raw را مطالعه کنید.

عنوان: ASP.NET MVC #8
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۱۳ ۲۲:۱۰:۰۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

معرفی HTML Helpers

یک HTML Helper تنها 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 img و input استفاده کرد. یا به کمک HTML Helpers 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 همچنین Event model نیستند.

ASP.NET MVC به همراه تعدادی متد HTML Helper توکار است و برای دسترسی به آن‌ها شیء Html که وهله‌ای 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 متد الحاقی باید پیوست شود و سپس از طریق شیء Html می‌توان به این متد 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 encoded به مرورگر تحویل می‌دهد. برای تغییر این رویه باید اندکی متد 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 Razor آن‌را encode نکند.

تعریف HTML Helpers سفارشی به صورت عمومی:

می‌توان فضای نام MvcApplication4.Helpers این مثال را عمومی کرد. یعنی بجای اینکه بخواهیم در هر View آن‌را ابتدا تعریف کنیم، یکبار آن‌را همانند تعاریف اصلی یک برنامه ASP.NET MVC، عمومی معرفی می‌کنیم. 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 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 Viewهای برنامه در دسترس خواهند 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 آن‌را برای بازنویسی متد 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 HTML Helpers ویژه، به کمک امکانات Razor

نوع دیگری از این متدهای کمکی، Declarative HTML Helpers نام دارند. از این جهت هم Declarative نامیده شده‌اند که مستقیما درون فایل‌های cshtml یا vbhtml به کمک امکانات Razor قابل تعریف هستند. تولید این نوع متدهای کمکی به این شکل نسبت به مثلا روش TagBuilder ساده‌تر است، چون توسط Razor به سادگی و به نحو طبیعی‌تری می‌توان تگ‌های HTML و کدهای مورد نظر 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 ایی 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 (دقیقا 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 یافته و مثلا نام ProductsList.cshtml 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 هستند 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 Viewها 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 مستقیما داخل یک View هم تعریف کرد. بدیهی است در این حالت قابلیت استفاده مجدد از آن‌را به همراه داشتن Viewهایی تمیز و کم حجم، از دست خواهیم داد.

جهت تکمیل بحث

Turn your Razor helpers into reusable libraries

نظرات خوانندگان

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

بسیار متشکر آقای نصیری ، به شدت این سلسله مباحث را دنبال می کنم. خداقوت.

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

مرسی خیلی عالیه لطفا ادامه بدید...

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

استاد وحید خان نصیری
بسیار ممنون از مطالب عالیتون

نویسنده: 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 و در مورد MvcHtmlString.cteate بدهید. قبلا از بابت پاسخ تشکر میکنم.

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

در قسمت پنجم توضیح دادم.

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

سلام
در متد الحاقی "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 ؟

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

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

نویسنده: 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 وهله‌ای از کلاس HtmlString است. چون محصور سازی شده، MSDN نوشته Html-encoded که باید می‌نوشت «Wraps the string in a class that implements IHtmlString»

ارائه یک رشته به صورت وهله‌ای 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());
}
```

و این هم سورس MvcHtmlString :

```
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 HtmlString : MvcHtmlString یعنی MvcHtmlString از جنس همان HtmlString ایی است که متدهای MvcHtmlString.Create و Html.Raw باز می‌گردانند.

کاربرد MvcHtmlString عموما در بازگرداندن مقادیر از helper methods است؛ زمانیکه می‌دانید خروجی آن دارای تگ‌های html است. مثلا قرار است یک دکمه را نمایش دهد. به این ترتیب خروجی متد از 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: ۱۳۹۱/۱۰/۱۴ ۱۷:۲

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 داره؟

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

<h2>Index</h2>

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

یا

@ProductsList.GetProductsList(MvcApplication4.Models.Products)
```

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

همان اصول زبان سی شارپ اینجا هم برقرار است. آیا در حالت متداول 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 سی‌شارپ باید رعایت شود:

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

در اینجا وهله یا instance ایی باید به آن پاس شود نه syntax آن و ... @Model یک وهله است.

نویسنده: جواد
تاریخ: ۱۳۹۱/۱۰/۱۴ ۱۸:۱۷

ممنون

فقط یه نکته هنوز برام مشخص نشده که تو کدوم قسمت وهله 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 وهله ای از

لیست محصولات(که همون Products هست) به حساب میاد، درسته؟ پس فکر کنم مشکل من بر میگرده به اینکه با ساختار List<T>class آشنایی ندارم ، به این خاطره یا چیز دیگس؟

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

سورس کامل مثال‌های این سری رو دریافت کنید: MVC_Samples
جایی که وهله‌ای از اشیاء به View متناظر ارسال می‌شود در اکشن متد مشخص شده است (return View):

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

نویسنده: جواد
تاریخ: ۱۳۹۱/۱۰/۱۴ ۱۹:۵۲

یعنی @Model یه viewdatadictionary هست... که با این خط کد

```
return view(products)
```

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

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

در اینجا Model یک خاصیت از شیء ViewData است که در ControllerBase تعریف شده .
مراجعه کنید به قسمت 5 برای توضیحات بیشتر « بررسی نحوه انتقال اطلاعات از یک کنترلر به Viewهای مرتبط با آن »

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

« *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 نمائید: »

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

نویسنده: vici
تاریخ: ۱۳۹۲/۱۰/۲۷ ۱۹:۶

سلام؛ چرا داخل html helper نمی‌تونیم از Actionlink استفاده کنیم؟ هدفم ایجاد لینک هایی ست که به یه متد(Detail) اشاره کنن همراه با Id مورد نظر برای دیدن جزییات. ممنون

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

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 ۱۳:۹

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 ۱۳:۳۶

متد 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 ۱۵:۳۸

```
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*
MvcApplication1.Models.Products    Products()
using System.Collections.Generic;

namespace MvcApplication1.Models
{
    2 references
    public class Products : List<Product>
    {
        1 reference
        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

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، در فایل Index.cshtml، نحوه‌ی تعریف نوع مدل را توسط @model و استفاده از مقدار آن‌را 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، مثلا توی یک فولدر، چه جوری باید ازش استفاده کرد؟

نویسنده: وحید نصیری

تاریخ: ۱۳۹۲/۱۲/۱۳ ۱۳:۲۳

از razor generator می‌توانید استفاده کنید.

نویسنده: پاییز
تاریخ: ۱۳۹۳/۱۲/۱۷ ۱۲:۲۴

در ورژن کنونی ASP.NET MVC دسترسی به کلاس HtmlHelper در یک Declarative Html Helper وجود ندارد. علاوه بر روش هایی که عنوان شد، می‌توانید از فراخوانی مستقیم متد ActionLink توسط کلاسی که آن را بسط می‌دهد استفاده کنید.

مثال:

```
@using WebApplication1.Models
@helper GetProductsList(List<Product> products)
{
    <ul>
        @foreach (var item in products)
        {
            <li>
                @System.Web.Mvc.Html.LinkExtensions.ActionLink(

((System.Web.Mvc.WebViewPage)CurrentPage).Html,
                                                                item.Name,
                                                                "ActionName",
                                                                "ControllerName",
                                                                new { id = item.ProductNumber },
                                                                null)
            </li>
        }
    </ul>
}
```

بدیهی است در اینجا به جای فراخوانی به صورت زیر

```
Html.ActionLink("foo")
```

این گونه فراخوانی کرده ایم:

```
ActionLink(Html, "foo")
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۲/۱۷ ۱۲:۵۰

در قسمت بعدی ، بحث متد @ helper GetProductsList(**WebViewPage** page, List<Product> products) را مطالعه کنید.

نویسنده: پاییز
تاریخ: ۱۳۹۳/۱۲/۱۷ ۱۴:۴۳

متشکرم
راه حلی که در قسمت بعدی ارائه شده، بهتر و ساده‌تر است.

در مجموع شاید بهتر باشه در ورژن‌های آتی ASP.NET MVC، تیم توسعه دهنده راهی ساده‌تر برای دسترسی به HtmlHelperهای توکار در Declarative Html Helper ها ارائه کند.

کجا می‌توان این موضوع را بهشون گزارش داد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۲/۱۷ ۱۵:۱

تا نگارش 5 در اینجا . از نگارش 6 به بعد در اینجا .

نویسنده: پاییز
تاریخ: ۱۳۹۳/۱۲/۱۹ ۱۳:۹

درخواست این موضوع را دادم.
جواب دادند که در ورژن بعدی، فعلا پیاده سازی برای این موضوع انجام نشده و به جای آن فیوچری به نام Tag Helpers در ورژن بعدی ارائه خواهد شد.

عنوان: ASP.NET MVC #9
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۱۵ ۱۳:۳۶:۰۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

مروری 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 control، GridView 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 HTMLها را باید خودمان تولید کنیم. البته در عمل خیر. یک نمونه از آن‌را در قسمت قبل مشاهده کردیم که چطور می‌توان منطق تولید تگ‌های HTML را کپسوله سازی کرد و بارها مورد استفاده قرار داد. به علاوه فریم ورک 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.
یک مثال:

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

در اینجا از متدی به نام ActionLink استفاده شده است. شیء Html هم وهله‌ای از کلاس HtmlHelper است که در تمام Viewها قابل دسترسی می‌باشد.
در این متد، 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 متد یا اکشنی در آن است که فراخوانی خواهد شد (البته هر کدام از این HtmlHelperها به همراه تعداد قابل توجهی overload هم هستند).
زمانیکه این صفحه را رندر کنیم، به خروجی زیر خواهیم رسید:

```
<a href="/About">About us</a>
```

در این لینک نهایی خبری از متد Index ایی که معرفی کردیم، نیست. چرا؟
متد ActionLink بر اساس تعاریف پیش فرض مسیریابی برنامه، سعی می‌کند بهترین خروجی را ارائه دهد. مطابق تعاریف پیش فرض برنامه، متد Index، اکشن پیش فرض کنترلرهای برنامه است. بنابراین ضرورتی به ذکر آن ندیده 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 ویژه، 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 متد جدیدی به نام 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 متد ActionLink بکارگرفته 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 فریم ورک واگذار 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، متد 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 متد Details، ابتدا ProductNumber دریافت شده و سپس شیء محصول متناظر با آن، به View این متد، 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 از نوع Product را قرار است به View ارسال کنیم (مانند مثال قسمت پنجم)، می‌توان در صفحه باز شده تیک Create a strongly typed view را گذاشت و سپس Model class را از نوع Product انتخاب کرد و در قسمت Scaffold template هم Details را انتخاب نمود. به این ترتیب Code generator توکار VS.NET 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 فوق را در پوشه App_Code، همانند قسمت قبل قرار دهیم، به متد Html.ActionLink 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 وهله‌ای از شیء System.Web.WebPages.Html.HtmlHelper، به دنبال متد ActionLink 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 WebViewPage 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>
        }
    </ul>
}
```

سپس برای استفاده از آن در یک View خواهیم داشت:

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

متد ActionLink و عبارات فارسی

متد ActionLink آدرس‌های وبی را که تولید می‌کند، URL encoded هستند. برای نمونه اگر رشته‌ای که قرار است به عنوان پارامتر به اکشن متد ما ارسال شود، مساوی Hello World است، آن‌را به صورت Hello%20World 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 مثلا «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 متد ارسال می‌گردد.
مرورگرهایی مثل فایرفاکس و کروم، مشکلی با نمایش این لینک به شکل اصلی فارسی آن ندارند (حین نمایش، URL decoding را اعمال می‌کنند). اما اگر مرورگر مثلا IE8 باشد، کاربر دقیقا 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 حین نمایش نهایی آن، آن‌را فارسی نشان خواهد داد. حتی زمانیکه کاربر بر روی آن کلیک کند، به صورت خودکار کاراکترهایی 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 دقیقا همین رفتار را در مورد لینک‌های داخل صفحه دارند).
خلاصه این توضیحات متد کمکی 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 http://site/blog/details/id/text. 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. عموما 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 Redirectایی اینجا صورت نمی‌گیره، میشه مثلا 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 نباشید. 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 queryString تبدیل می‌شود . 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. route 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: ۱۳۹۱/۰۴/۰۵ ۱۲:۴۲

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: ۱۳۹۱/۰۴/۰۵ ۱۲:۵۲

به قسمت چهارم این سری برای توضیحات بیشتر مراجعه کنید.

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

سلام
ممنون از راهنماییتون . به پارامتر دوم متد 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 .

ممنون

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

خیر. علت این است که در MVC کارها از یک اکشن متد 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 متد یا نقطه آغازین کار، معنی ندارد.

نویسنده: امیر
تاریخ: ۱۳۹۱/۰۵/۲۷ ۱۷:۱

سلام
طراحی viewها کمی به نظر سخت میاد چون امکان هیچگونه preview برای برنامه نویس فراهم نیست
راه حلی برای این مشکل وجود داره یا خیر؟

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

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

نویسنده: حمید
تاریخ: ۱۳۹۱/۰۷/۰۷ ۱۳:۵۸

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

با سلام وتشکر 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 پروسه 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 پروپرتی‌ها زیاد باشد باعث Scroll خوردن میشود.
ممنون.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۲۳ ۱:۷

نیاز به استفاده از یک CSS Framework خواهید داشت. مثلا: « معرفی فریم ورک 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: ۱۳۹۱/۱۰/۱۵ ۹:۱۳

Post در اینجا کنترلر است. مراجعه کنید به قسمت چهارم ، بحث تنظیمات کنترلر و همچنین اکشن متد پیش فرض تعریف شده در global.asax.cs

نویسنده: جواد
تاریخ: ۱۳۹۱/۱۰/۱۵ ۱۲:۳۷

برای نمایش مقاله مورد نظر در صفحه مگه از یک id برای شناسایی اون استفاده نمیشه... 812 رو که من دستی تغییر میدم مقاله عوض میشه و مقاله مربوط به اون نمایش داده میشه پس اینطور نتیجه گرفتم که 812 آی دی و asp-net-mvc-9 بایستی text باشه چون تاثیری در آدرس نداره... میمونه post که کنترلر هست... پس اینجا اکشن نداریم... اگه میشه لطفا توضیح بدید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۰/۱۵ ۱۳:۲۸

```
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 آن) که در قسمت چهارم بحث شد ، اگر نام کنترلری وارد نشود، از نام پیش فرض Home استفاده می‌شود (مثلا برای مدیریت آدرس‌های ریشه سایت http://localhost و یا http://site.com). اگر جایی در آدرسی، نام اکشن متدی ذکر نشود، از نام متد پیش فرضی به نام Index استفاده می‌شود و الی آخر.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۱/۰۷ ۱۶:۴۶

یک نکته تکمیلی یا یک ... تجربه:
منهای موتورهای جستجوی خوب در اینترنت، مابقی قدرت پردازش لینک‌های یونیکد را ندارند. به همین جهت از متد EmitCleanUnicodeUrl استفاده نکنید. مهم نیست که این لینک‌ها در IE شکل زیبایی نخواهند داشت، مهم این است که تعداد خطاهای لاگ شده در برنامه شما در اثر عدم قدرت پردازش لینک‌های یونیکد توسط بسیاری از ربات‌های متفرقه به حداقل می‌رسد.

نویسنده: امیرحسین
تاریخ: ۱۳۹۲/۱۱/۰۸ ۵:۱۰

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
{
    <ul>
        @foreach (var item in products)
        {
            <li>@EmitCleanUnicodeUrl(@page.Html.Action
        }
    </ul>
}
```

NullReferenceException was unhandled by user code

Object reference not set to an instance of an object.

Troubleshooting tips:

- Check to determine if the object is null before calling the method.
- Use the "new" keyword to create an object instance.
- Get general help for this exception.

Search for more Help Online...

Exception settings:

☑ Break when this exception type is user-unhandled

Actions:

View Detail...

Enable editing

Copy exception detail to the clipboard

Open exception settings

ولی وقتی کد رو در view میذارم (مانند مثال مطلب جاری) دیگه خطا نمیده.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۰۸ ۹:۴۳

همانطور که در مقاله فوق هم ذکر شده، اگر محل فایل helper شما در پوشه App_Code است (و نه یک View قرار گرفته در پوشه Views برنامه)، نمی‌توانید مستقیما با Html@ها کار کنید. نیاز است از طریق پارامتر WebViewPage page به آن‌ها دسترسی پیدا کنید؛ یعنی:

```
@page.Html.Raw(data)
```

عنوان: ASP.NET MVC #10
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۱۷ ۰۰:۳۰:۰۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

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 رویدادگردان 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 متدی در آن نمی‌دانند که الان بر روی View آن، یک گرید قرار دارد یا یک دکمه یا اصلا 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 مانند HttpContext، Request، RouteData 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 لاگین ساده را طراحی کنیم تا بتوانیم هر سه حالت ذکر شده فوق را در عمل بررسی نمائیم. بحث HTML Helpers 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 متد 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 لاگین ظاهر شود.

کدهای کامل کنترلر لاگین 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 لاگین برنامه:

در View صفحه لاگین 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 متد کمکی Html.BeginForm انجام می‌شود. در اینجا برای مثال می‌شود یک فرم را به کنترلری خاص و متدی مشخص در آن نگاشت نمائیم.

از عبارت using هم برای درج خودکار تگ بسته شدن فرم، در حین dispose شیء 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 For آن‌ها و همچنین lambda expression ایی که بکارگرفته شده برای چیست؟

متدهای کمکی Html.TextBox و Html.Password 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 لاگین برنامه، به سه متد متفاوت قرار گرفته در کنترلری به نام Login ارسال خواهند 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 لاگین 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 HttpGet و HttpPost در مورد نحوه دسترسی به این متدها، محدودیت قائل شده‌ایم. به این معنا که تنها در حالت Post است که متد LoginResult در دسترس خواهد بود و اگر شخصی نام این متدها را مستقیما 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 متد استفاده می‌شد 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 مثلا 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 ایی به این شکل، در حالت پیش فرض، محدودیت اجرایی HttpGet را دارد. بنابراین احتمال اجرا شدن آن بالا است. اما زمانیکه متد delete را به HttpPost محدود کردید، دیگر این نوع حملات جواب نخواهند داد و حتما نیاز خواهد بود تا اطلاعاتی به سرور Post شود و نه یک Get ساده (مثلا کلیک بر روی یک لینک معمولی)، کار حذف را انجام دهد.

توسط ActionName می‌توان نام دیگری را صرفنظر از نام متد تعریف شده در کنترلر، به آن متد انتساب داد که توسط فریم ورک در حین پردازش نهایی مورد استفاده قرار خواهد گرفت. برای مثال در اینجا به متد LoginResult دوم، نام LoginResultWithParams را انتساب داده‌ایم که در فرم دوم تعریف شده در View لاگین برنامه مورد استفاده قرار گرفته است.
وجود این ActionName هم در مثال فوق ضروری است. از آنجائیکه دو متد 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 لاگین برنامه، چهار breakpoint را تعریف کنید. سپس برنامه را در حالت دیباگ اجرا نمائید و تک تک فرم‌ها را یکبار با کلیک بر روی دکمه لاگین، به سرور ارسال نمائید.

بررسی سه روش دریافت اطلاعات از کاربر در ASP.NET MVC

الف) استفاده از اشیاء Context

در ویژوال استودیو، در کنترلر لاگین 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 خواهند 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 متد LoginResult می‌توانید 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 شبیه سازی 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 متد به صورت توکار اعمال می‌شود که باید به آن دقت داشت:

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 زیرهستند:

```
Request.Form
Request.QueryString
Request.Files
RouteData.Values
```

برای نمونه در متدی که با نام LoginResultWithParams مشخص شده، چون نام‌های دو پارامتر آن، با نام‌های بکارگرفته شده در Html.TextBoxFor و Html.PasswordFor یکی هستند، با مقادیر ارسالی آن‌ها مقدار دهی شده و سپس در متد قابل استفاده خواهند بود. در پشت صحنه هم از همان رکوردهای موجود در Request.Form (یا سایر موارد ذکر شده)، استفاده می‌شود. در اینجا هر رکورد مثلا 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 (مثلا 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 متد LoginResult مساوی string name است، اما نام

خاصیت تعریف شده در کلاس Account مساوی Name بود.

ج) استفاده از ویژگی جدیدی به نام Data Model Binding

در ASP.NET MVC چون می‌توان با یک Strongly typed view کار کرد، خود فریم ورک این قابلیت را دارد که اطلاعات ارسالی یکی فرم را به صورت خودکار به یک وهله از 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 کاملا 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 return 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 دیباگ برنامه، بررسی نمائید.
بنابراین تفاوتی نمی‌کند که از چندین پارامتر استفاده کنید یا اینکه کلا یک شیء را به عنوان پارامتر معرفی نمائید. 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 ASP.NET MVC 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 آن‌را مشاهده خواهید نمود.
3) HTTP posted file base model binder که توضیحات آن به قسمت بعدی موکول 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 هم یک نوع model binder ساده است) و اساسا 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، اصلا 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 binderها 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.
-1تو بعضی مثال ها بجای استفاده از [ActionName] از FormCollection بعنوان یکی از پارامترهای اکشن متد استفاده میشه(اگه اشتباه نکنم Scaffold هم از این روش استفاده میکنه).برای داشتن متد هم نام کدوم روش پیشنهاد میشه؟

-2همونطور 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 ActionName استفاده 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 مورد 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 است و نمیشه Viewایی رو از یک FormCollection استخراج کرد.
- هدف من تکمیل بحث بود. آشنایی با انواع روش‌های 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 کنترل‌های وب فرم‌ها به صورت پویا به صفحه اضافه بشن، دیگه eventها جهت دریافت مقادیر اون‌ها معنا نخواهند 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 Request.Form برای دریافت مقادیر کنترل‌های 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 برا ما اکشن متدها و View های CURD رو میسازه) بود.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۱۷ ۱۶:۰۳:۲۴

البته این Viewها (ی create/update و غیره) توسط scaffold template صفحه اضافه کردن view هم از روی یک مدل قابل تولید هستند (در لیست قالب‌ها هست). یعنی الزاما نیازی به EF نیست. چون EF مدل‌هاش دردسترس 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 عزیز
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 اید؟
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۱۸ ۰۱:۱۶:۵۶

عنوان شد که اگر از FormCollection به عنوان پارامتر استفاده بشه، می‌شود دو متد هم نام را در یک کنترلر داشت. پاسخ این است که خیر. تحت هر حالتی نمی‌تونید دو متد هم 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 متد هم نام درست کنید. یکی با پارامتر و یکی بدون پارامتر. هر دو هم httppost. بعد سعی کنید یک فرم را به سرور ارسال کنید. با پیغام controller type is ambiguous مواجه خواهید شد. روش حل این مساله استفاده از ActionName است.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۱۸ ۰۱:۳۸:۲۵

جهت تکمیل: می‌توان دو یا چند متد همنام در یک کنترلر داشت اگر از Verbهای مختلفی استفاده کنند. Post و Get و امثال این، Http Verb هستند.

نویسنده: محمد صاحب
تاریخ: ۱۳۹۱/۰۱/۱۸ ۱۴:۵۶:۰۱

کد کامپایل نمیشه +

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۱۸ ۱۶:۲۶:۰۲

بله. ویژگی‌های HttpPost و HttpGet مقدم نیستند بر اصول زبان مورد استفاده. اصول زبان مورد استفاده حین تعریف امضای متدها باید رعایت شوند (هر overloadایی رو نمیشه تعریف کرد). اما ... اگر ... این اصول رعایت شوند و کد شما کامپایل شود، آنگاه می‌توان دو یا چند overload را با verbهای مختلفی بدون مشکل دریافت خطای ambiguous استفاده کرد.

نویسنده: Mohsen
تاریخ: ۱۳۹۱/۰۱/۱۹ ۱۴:۴۷:۰۱

پس میشه گفت در واقع فریم ورک هر وقت با پارامتر (از نوع شی یا هر چیزی) روبرو می شود هم UpdateModel را فراخوانی می کند؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۱۹ ۱۶:۱۴:۲۶

سورس MVC3 در دسترس است: (^)
روشی که در متد TryUpdateModel بکارگرفته شده (در فایل Controller.cs) که مبتنی بر کار با اینترفیس 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 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 ۱۶:۴۲

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: ۱۳۹۳/۰۱/۰۱ ۱:۵۹

وقتی method فرم رو روی Get تنظیم میکنم مقدار متغییرها 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: ۱۳۹۳/۰۱/۰۱ ۹:۲۴

برای فرم لاگین 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 لاگین 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 این‌کار 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: ۱۳۹۳/۰۸/۰۹ 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 میشن 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 دادید ننوشتید؟ چون من علاقه زیادی به استفاده از اشیا و کنترل‌های سروری ندارم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۸/۱۰ 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@ها سبب خواهد شد، اگر نام خاصیتی را در کدهای خود تغییر دادید، بتوان پیش از اجرای سایت و در حین کامپایل، دقیقا تشخیص داد که کدام Viewهای یک برنامه‌ی بزرگ دیگر کار نخواهند کرد. همچنین این HTML helperها، 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: سعیدجلالی
تاریخ: ۱۳۹۳/۰۹/۰۱ ۱۴:۳۲

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 فیلدهای decimal پر شده ولی نتیجه bind نمیشه
همین فیلدهای decimal رو اگر با اعداد صحیح نه اعشاری پر کنم binding انجام میشود!
مشکل از کجا است؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۰۱ ۱۴:۵۲

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 شده‌است.

عنوان: ASP.NET MVC #11
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۱۹ ۰۹:۵۲:۰۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

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 نمائید:

```
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 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 حین model binding خودکار به آن توجه نشود، سایت را با مشکلات حاد امنیتی مواجه خواهد کرد. سیستم پیشرفته است. می‌تواند به صورت خودکار ورودی‌های کاربر را تبدیل به یک شیء حاضر و آماده کند ... اما باید حین 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 نمائید. 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 مرتبط با 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 NH یا حتی ADO.NET کلاسیک و یا از Micro ORM هایی که پس از ارائه دات نت 4 مرسوم شده‌اند. تهیه یک ToList یا Insert و Update با این فریم ورک‌ها 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 درنظر 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 کاملا 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 create a strongly typed view را انتخاب کرده و سپس از لیست مدل‌ها، User را انتخاب نمائید. Scaffold template را هم بر روی حالت List قرار دهید.
برای متد Details هم به همین نحو عمل نمائید.
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 Scaffold template 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 کد را در اندک زمانی می‌توان تولید کرد. بدیهی است حتما نیاز به سفارشی سازی کدهای تولیدی وجود خواهد داشت. مثلا شاید نیازی نباشد فیلد پسود 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 حتما 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. متد Delete هم 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 قرار می‌دهیم 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.

علت وجود دو متد، به ازای هر Edit یا Create

به ازای هر کدام از متدهای Edit و Create دو متد HttpGet و HttpPost را ایجاد کرده‌ایم. کار متدهای HttpGet نمایش Viewهای 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 Http 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 در متد Create قرار دهید. سپس به صفحه ایجاد کاربر جدید مراجعه کرده و مثلا 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 حتما 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، مجددا 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 مجددا 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 دسکتاپ 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 ASP.NET MVC بجای استفاده از ترفندی به نام ViewState، مجددا اطلاعات همان مدل متناظر با 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 مشخص است، View ما هم از نوع 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 مجددا همان 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» مجدا 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 ... اصلا قابل قبول نیست و فوق‌العاده 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 خطاها، در متد 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)
```

EditorFor سعی می‌کند اندکی هوش به خرج دهد. یعنی اگر خاصیت دریافتی مثلا از نوع bool بود، خودش یک checkbox را در صفحه نمایش می‌دهد. همچنین بر اساس متادیتا یک خاصیت نیز می‌تواند تصمیم گیری را انجام دهد. این متادیتا منظور attributes و data annotations ایی است که به خواص یک مدل اعمال می‌شود. مثلا اگر ویژگی HiddenInput را به یک خاصیت اعمال کنیم، به شکل یک فیلد مخفی در صفحه ظاهر خواهد شد.
یا متد Html.DisplayFor، اطلاعات را به صورت فقط خواندنی نمایش می‌دهد. اصطلاحا به این نوع متدها، Templated Helpers هم گفته می‌شود. بحث بیشتر درباره‌ای این موارد به قسمتی مجزا و مستقل موکول 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 متد ValidationMessageFor کار نمایش خطاهای اعتبارسنجی مرتبط با یک خاصیت مشخص را انجام می‌دهد. بنابراین اگر قصد ارائه خطایی سفارشی و مخصوص یک فیلد مشخص را داشتید، در متد 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 متد 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 پیشرفته جدید ناآگاهانه مورد استفاده قرار گیرد می‌تواند منشاء حملات ناگواری شود که به نام «Mass Assignment» شهرت یافته‌اند.
همان صفحه ویرایش اطلاعات را درنظر بگیرید. چک باکس IsAdmin قرار است در قسمت مدیریتی برنامه تنظیم شود. اگر کاربری نیاز داشته باشد اطلاعات خودش را ویرایش کند، مثلا پسوردش را تغییر دهد، با یک صفحه ساده کلمه عبور قبلی را وارد کنید و دوبار کلمه عبور جدید را نیز وارد نمائید، 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 چندین 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 خواهند 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، متد edit را به شکل زیر نیز می‌توان بازنویسی کرد. در اینجا متدهای توکار UpdateModel و TryUpdateModel 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 متدی خاصیت IsAdmin را مجددا 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 آشنا است؛ اما در اینجا مفهوم متفاوتی دارد. در الگوی MVVM، کلاس‌های ViewModel شبیه به کنترلرها 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 Domain Model و View 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 ساختار View model را دیکته می‌کند و نه کنترلر.

ج) View modelها صرفا 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 الزامی هم ندارد که این اطلاعات مستقیما به domain models مرتبط شوند. برای مثال اگر قرار است firstName+LastName در View 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 مثلا 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.

ViewModel چگونه پیاده سازی می‌شود؟

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 مثلا دو Model نیاز دارد. این‌ها را به این شکل محصور و کپسوله می‌کنیم. اگر View، واقعا به تمام فیلدهای این کلاس‌ها نیاز داشته باشد، این روش صحیح است. در غیر اینصورت، این روش نادرست است (و متاسفانه همه جا هم دقیقا به این شکل تبلیغ می‌شود).

ViewModel محصور کننده یک یا چند مدل نیست. در اینجا حس غلط کار کردن با یک ViewModel را داریم. ViewModel 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. مثلا نمایش نام استان‌ها که نهایتا Id انتخابی آن قرار است در برنامه استفاده شود.

خلاصه علت وجودی ViewModel این موارد است:

الف) Model برنامه را مستقیما 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 (مثلا تعاریف عناصر یک دراپ داون 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 IsAdming 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 متد به 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");
}
```

View 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 enctype فرم آپلود، مورد استفاده قرار گرفته است. به عبارتی هر جایی که مشخص نبوده چه تعداد ویژگی یا کلا چه ویژگی‌ها و خاصیت‌هایی را می‌توان تنظیم کرد، اجازه تعریف آن‌ها را به صورت anonymously typed objects میسر کرده‌اند. یک نمونه دیگر آن در متد routes.MapRoute فایل Global.asax.cs است که پارامتر سوم دریافت مقدار پیش فرض‌ها نیز anonymously typed object است. یا نمونه دیگر آن‌را در همین قسمت در جایی که لینک delete را به فرم تبدیل کردیم مشاهده نمودید. مقدار routeValues هم یک anonymously typed object معرفی شد.

سفارشی سازی model binder پیش فرض ASP.NET MVC

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 نهایتا 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 اینترفیس 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
            {
```

```
                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;
        }
    }
}
```

سپس برای معرفی PersianDateModelBinder جدید تنها کافی است سطر زیر را

```
ModelBinders.Binders.Add(typeof(DateTime), new PersianDateModelBinder());
```

به متد 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 model binder 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 آن‌را 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 ۱۱:۱۱:۰۲

آموزش ها واقعا عالیه. ممنون از شما
فقط اگه میشه در مورد UpdateModel و TryUpdateModel یکم بیشتر توضیح بدید که کارکردشو 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 (InvalidOperationException) اما TryUpdateModel خیر و فرصت خواهید داشت تا با بررسی نتیجه اعتبار سنجی، تصمیم بهتری را اتخاذ کنید.
کلا هرجایی در دات نت Try دیدید، هدفش همین مساله است. مثلا int.Parse داریم و int.TryParse. در مورد سایر نوع‌های عددی هم این دو نوع متد وجود دارند. متد Parse اگر نتواند تبدیل را انجام دهد، یک استثناء را صادر می‌کند، اما متد 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" viewmodel بدون وابستگی به model"هست که شما فرمودین . ولی تو این مثال isAdmin من متوجه نشدم که چه مشکلی پیش میاد. به فرض کاربر به طور جعلی به isadmin مقدار بده، مگه متد post برای changePassword با متد post برایِ Edit متفاوت نخواهد بود؟ پس مقدار جعلی isAdmin چه تاثیری میخواد بذاره.
درضمن 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 model کامل 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

واقعا 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 می گیرند. اما به نظر من استفاده از ViewModel ها در MVC صحیح نیست. دلیل اصلی اش اینه که ساختار MVC و اهداف اون رو تحت تأثیر قرار می ده. بقیه دلایل هم نهایتاً به همین بر می گردند.

معمولا ظاهر نرم افزارهای کاربردی تحت وب بیشتر از منطق درونی آنها تغییر می کنند. مثلا برای تنوع، یک کار ثابت رو به شکل های جدید انجام می دهند. در واقع برای یک کاری که توسط کنترلر بر روی مدل انجام می گیرد، view های جدید درست می شود. 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 لایه ها می شود.

دلیل محبوبیت ViewModel ها همانا سادگی آن ها و راحتی طراحی است. یعنی در یک صفحه نسبتاً پیچیده که قسمت های مختلفی دارد، خیلی زود می شه یک ViewModel درست کرد و همه قسمت های 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 حل های جایگزین فکر کرد. به عنوان نمونه:

1- می شه از امکانات خود MVC برای انتقال اطلاعات بین کنترلر و ویو استفاده کرد. می تونیم نمونه ای از مدل رو به ویو بفرستیم و برای موارد اضافی از ViewBag استفاده کنیم. مثلاً می شه اسامی استان ها رو که به یک DropDownList بایند خواهند شد، به شکل زیر بفرستیم:

```
ViewBag.StatesList = new SelectList(_bll.GetStatesList(), "Id", "Title"); //ltr
```

2- این توضیح رو بدم که MVC ما رو تشویق می کنه که منطق نرم افزارمون رو به ماژول های کوچک تقسیم کنیم و هر ماژول رو به تنهایی مدیریت کنیم. همه جا این تقسیم بندی ها هست. اول کل ساختار نرم افزار رو به سه لایه M و V و C تقسیم می کنه. بعد View ها و کنترلر های هر مدل رو که (که در عالم واقعیت هم مستقل هستند) از هم تفکیک می کنه و یاد میده اینها هر کدوم باید تو فولدر جداگانه ای باشند. Area ها هم در همین راستا هستند. حتی در داخل کنترلر ها، هر کدوم از اکشن ها یک کار واحد کوچک رو انجام می دهند. اما ViewModel این ساختار رو می شکنه و ما رو مجبور می کنه 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 بدیم. مثلاً ممکنه در یک صفحه واحد، هم لیست دانش آموز ها رو به همراه امکانات حذف، اضافه و ویرایش داشته باشیم و هم لیست کلاس ها رو. علاوه بر اون یک فرم ارسال نظر در پایین صفحه، یک textbox برای فیلتر کردن و ... داشته باشیم. درسته که می شه 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 کدوم از اینها مستقلاً پیاده سازی شوند. مثلاً صفحه ای برای دانش آموزان، صفحه ای برای کلاس ها و ... غصه اون صفحه بزرگ رو هم نخورید. می شه هر کدوم 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 کدوم از این قطعات در قسمت های مختلف قابل استفاده می شوند. به این ترتیب می توانیم با جابجایی آن ها در صفحات مختلف، ظاهر نرم افزارمون رو به دلخواه تغییر بدیم بدون اینکه مجبور باشیم تغییری در کنترلر ها و قسمت های دیگه ایجاد کنیم.

در کل ViewModel ها استقلال لایه ها را کاهش می دهند، شدیداً موجب افزایش کدها می شوند، قابلیت گسترش منطقی ندارند و رفته رفته بزرگتر و بزرگتر می شوند، ماژولار 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
تاریخ: ۱۳۹۱/۰۱/۲۰ ۱۶:۲۱:۵۳

خلاصه موردی را که عنوان کردید این است:

اگر یک صفحه داریم که از مثلا 4 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 viewهای مختلفی برای مثال خرد کرد. اما نهایتا هر کدام از این اجزای کلی سیستم اصلی، نیاز به ViewModel دارند که این مورد، بحث اصلی جاری است. یعنی تفاوت قائل شدن بین domain model و view model.

پوشه‌ی مدلی که در ساختار پروژه پیش فرض ASP.NET MVC قرار داره در واقع امر یک ViewModel است و نه مدل به معنای تعریف مدل‌های domain سیستم چون قرار است خواص آن مستقیما در View مورد استفاده قرار گیرند.

استفاده از ViewModelها کار را اندکی بیشتر می‌کنند، چون نهایتا خواص آن‌ها باید به مدل اصلی نگاشت شوند اما خوشبختانه برای این مورد هم راه حل هست و روش پیشنهادی، استفاده از کتابخانه‌ی سورس بازی است به نام AutpoMapper : (^)

نویسنده: Salehi
تاریخ: ۱۳۹۱/۰۱/۲۰ ۲۳:۳۴:۵۱

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 strongly type بودن view 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: ۱۳۹۱/۰۱/۲۱ ۰۰:۲۱:۳۵

در مورد Refactoring فایل‌های View در قسمت بعدی توضیح دادم.

نویسنده: Ariyous Abdiyous
تاریخ: ۱۳۹۱/۰۱/۳۰ ۱۶:۴۲:۲۳

واقعا عالی بود. ممنون بابت مطلبتون

نویسنده: شهروز جعفری
تاریخ: ۱۳۹۱/۰۴/۰۲ ۱۷:۱۵

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 باتشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۰۲ ۱۷:۳۱

به شخصه برای تمام Viewهایی که درست می‌کنم از ViewModel استفاده می‌کنم و نام آن‌ها را هم نام view مورد نظر + کلمه ViewModel درنظر می‌گیرم.
به هیچ عنوان هم از یک مدل (domain model) به عنوان مدل یک View استفاده نمی‌کنم خصوصا 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 به نام 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 مستقیما در MVC استفاده شوند. این M در MVC به معنای مدلی مخصوص View است و نه کلاس‌های اصلی دومین شما.

نویسنده: شهروز جعفری
تاریخ: ۱۳۹۱/۰۴/۰۲ ۱۷:۴۷

حق با شماست ولی آیا به نظر شما این کار تکرار کد نویسیو باعث نمیشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۰۲ ۱۷:۵۲

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 برنخورید.
در کل انجام کار اصولی الزاما به معنای کمتر کدنوشتن نیست. ضمن اینکه کتابخانه auto-mapper کار نگاشت viewmodel به model رو می‌تونه 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 باید از viewmodelهای متفاوتی استفاده کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۰۳ ۹:۱۱

کاری که من انجام می‌دم، Refactor یک فرم به یک PartialView 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 auto-mapper اینو مپ کنم به viewmodel و تو ویو نمایش 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 مقادیر controller و 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 ؟
2- در تمامی مثالها از برای نمایش Viewها از Modelهای ساده استفاده شده بود . در صورتی که بخواهم در یک view از Modeli استفاده کنم که یک Property آن لیستی از یک Model دیگر باشد به چه صورت است ؟
آیا باید viewها را همیشه به Modelهای جدا تفکیک کنیم ؟
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 model که type فقط از نوع 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 صریحا مشخص نمی‌کنید، از نام متد فراخوان به عنوان view متناظر استفاده می‌گردد. در اینجا هم به همین نحو است. ارسال به کنترلر متناظر با View جاری و متد پیش فرضی که در routing تعریف شده است، صورت می‌گیرد. البته می‌شود تمام این‌ها را هم صریحا 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. مثال من یک مثال عمومی بود. شما بر اساس contextهای دریافتی در حین پیاده سازی یک binder می‌تونید اون رو تا حد امکان سفارشی سازی کنید که مثلا برای یک کنترلر خاص یا یک مدل خاص فقط عمل کند.

نویسنده: hasani
تاریخ: ۱۳۹۱/۰۵/۱۶ ۲۱:۴۰

با سلام و عرض خسته نباشید 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 Nullable باشد، با خطا موجه میشویم که برای رفع آن خط زیر هم باید به Application_Start 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 دوتا از این ویژگی‌هارو در مثالتون رعایت نکردین.
اول، گفتین هر ViewModel از یک View استفاده میکنه. که در مثالتون از خواص اضافی استفاده کردین. که شاید از دو یا چند View استفاده کرده است.
دوم، در متن گفتین در ViewModel منطقی درکار نیست. همچنین گفتین ViewModel از کنترلر ساختارشو نمیگیره و از View میگیره.
پس مثلا 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

- MyViewModel اول مطرح شده: عنوان شد که این مورد نباید استفاده شود.
MyViewModel دوم مطرح شده: این صحیح است و فیلدهایی که در آن مشاهده می‌کنید بر اساس ساختار View (ایی فرضی جهت توضیح بحث) تعریف شدن و تناظر یک به یک وجود دارد.
- بله ViewModel ساختار فیلدهای تشکیل دهنده‌اش رو (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 نهایتا در اختیار View قرار می‌گیره. بنابراین فرق است بین طراحی ساختار یک کلاس بر اساس Viewایی متناظر و در ادامه نحوه مقدار دهی آن در یک کنترلر و نهایتا 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. مثلا جمع زدن نام‌ها؛ تبدیل تاریخ، محاسبه درصد مالیات و موارد جزئی از این دست که مستقیما در View صرفا 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 propertyهای موجود در model در صورت هم نام بودن با کنترل‌های 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: ۱۳۹۱/۰۵/۲۸ ۲۳:۵۶

- مانند قسمت 10 این سری از FormCollection استفاده کنید.
- در قسمت 14 به این موضوع پرداخته شده.

نویسنده: محسن
تاریخ: ۱۳۹۱/۱۱/۲۳ ۱۴:۱۸

من یک مثال جهت استفاده از model و viewmodel می‌خواستم که چگونه قابل اجرا است

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۲۳ ۱۵:۶

قسمتی در سایت هست تحت عنوان نیازمندی‌های آموزشی . درخواست خودتون رو در آنجا مطرح کنید. اگر یکی از نویسندگان 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 می‌تونید به این جمع اضافه بشید. خارج از این، مثال و مطلب اضافه‌تری در دسترس من و ما نیست.

نویسنده: پیمان مهربانی
تاریخ: ۱۳۹۱/۱۲/۱۵ ۱۸:۵۵

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 ViewModelها باعث افزایش امنیت در مقابل بعضی از حملات و افزایش Performance با دوری از SELECT * میشه.
اما مطلبی که هنوز برای من روشن نشده رهایی از Duplicate شدن منطق اعتبارسنجی مدل هاست که به صورت Attribute در ViewModelها مشخص می‌شوند.و این تعریف Validation Ruleها سیستم رو دچار Duplicated Business می‌کنه.آیا راهی هست که تمامی Validation Ruleها در Model انجام شه و به صورت AutoMap بر روی ViewModelهای متناظر اعمال شود؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۱۵ ۲۱:۶

روشی در اینجا با استفاده از AutoMapper برای کپی کردن ویژگی‌ها از مدل‌ها به ViewModelها پیشنهاد شده.
+ امکان خارج کردن تعاریف ویژگی‌ها از یک کلاس توسط MetaDataType نیز پیش بینی شده. بین این صورت می‌شود تعاریف ویژگی‌های تعریف شده را به چند کلاس مختلف هم اعمال کرد.

نویسنده: نوید
تاریخ: ۱۳۹۱/۱۲/۲۶ ۱۱:۴۲

با سلام
هنگامی که از ViewModelها 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 بر قرار کرد.
در چند مثال دیدم که از یک تابع جداگانه برای نگاشت بین اطلاعات این کلاس‌ها استفاده میکردند ولی نحوه این نگاشت رو 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: ۱۳۹۱/۱۲/۲۶ ۱۲:۶

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 خطا ایجاد می‌شود چرا؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۱۱ ۲۲:۱۱

بررسی کنید که آیا مانند مثال آزمایش شده فوق، enctype 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 خطا ایجاد می‌شد) از طریق این پست حل شد .

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۳۱ ۹:۳۶

در مثالی که من در این قسمت مطرح کردم، نام کنترل ارسال فایل photo است، اما نام پارامتر اکشن متد متناظر 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 بخوام 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 ک من برای حل مشکل داشتم این بود که یک فیلد String در viewmodelقرار میدادم و مقادیر اونو وقتی اطلاعات از سمت کاربر میومد به مقدار میلادی تبدیل می‌کردم و در فیلد مربوطه قرار میدادم و وقتی اطلاعات رو از دیتابیس لود میکردم مقدار فیلد تاریخ رو به شمسی تبدیل می‌کردم و در فیلد string میذاشتم.)

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 کوئری‌های datetime توکار بانک‌های اطلاعاتی داره. اما ... این کوئری‌ها یکجا مشکل پیدا می‌کنند و آن هم گروه بندی بر اساس ماه‌های شمسی است که تطابقی با ماه‌های میلادی ندارند و اگر قرار باشد برای آن کوئری ویژه‌ای بر اساس datetime میلادی نوشت، به کوئری بسیار پیچیده و کندی خواهیم رسید. اما گروه بندی بر اساس ماه‌های مقادیر فیلد شمسی ذخیره شده بسیار سریع است (مثلا تهیه گزارش لیست خلاصه جمع حقوقی تمامی کارکنان در یکسال، گروه بندی شده بر اساس ماه‌های شمسی؛ ... آن‌هم فقط با نوشتن یک کوئری ساده و نه بیشتر).
- روش دیگر نوشتن یک TextBoxFor سفارشی است که تاریخ میلادی دریافت کند و شمسی نمایش دهد. بعد از ترکیب Model binder فوق استفاده شود برای دریافت مقدار نهایی آن.

نویسنده: لیلا
تاریخ: ۱۳۹۲/۰۹/۱۰ ۹:۲

با تشکر
در هنگام create مقدار ModelState.IsValid ، برابر با false است ایا راهکار این لینک ، راهکار مناسبی است؟

```
ModelState.Remove("Id");
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۱۰ ۹:۱۳

خیر. باید بررسی کنید که علت آن چه چیزی بوده:

```
string messages = string.Join("; ", ModelState.Values
                                        .SelectMany(x => x.Errors)
                                        .Select(x => x.ErrorMessage));
```

نویسنده: مصطفی
تاریخ: ۱۳۹۲/۱۰/۲۱ ۱۲:۱۲

سلام
دوستان وقتی روی لینک details کلیک می‌کنم این ارور رو میده:

The model item passed into the dictionary is of type 'mvc1.Models.User', but this dictionary requires a model item of type 'System.Collections.Generic.IEnumerable`1[mvc1.Models.User]'. به نظر شما از چی میتونه باشه
با تشکر فراوان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۰/۲۱ ۱۳:۳۲

- خروجی اکشن متد Details یک شیء User هست در اینجا. اما شما در View خودتان بجای شیء User، یک IEnumerable از آن‌را به عنوان نوع مدل تعریف کرده‌اید.
+ دریافت تمام مثال‌های MVC این سری: MVC_Samples

نویسنده: منصور
تاریخ: ۱۳۹۲/۱۱/۰۵ ۱۲:۹

سلام؛ ببخشید این ViewModel کجا تعریف میشن. مثلا اگر یک مدل اطلاعات کاربران داریم به سه تا فیلد نام و رمز عبور و IsAdmin. این ViewModel ی که شامل فقط فیلدهای نام و پسورد هست کجا و چطور تعریف میشن؟ ممنونم.

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۱/۰۵ ۱۳:۲۵

هرجا دوست داشتی. خواستی براش یک DLL جدا درست کن . خواستی داخل پوشه مدل‌های همین پروژه قرارش بده. استاندارد خاصی نداره.

نویسنده: kia
تاریخ: ۱۳۹۲/۱۱/۰۵ ۱۵:۵۵

با سلام؛ چند روز پیش متوجه وجود یک مشکل در فرم هایی که مقادیرشون به وسیله Ajax به سرور ارسال می‌شوند توی IE در

حالت Compatibility View شدم.

شرح مسئله :

View من به شکل زیره :

```
<% 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="لطفاً نام کاربری انتخابی خود را وارد کنید" data-toggle="tooltip" data-placement="top">
                                        <button id="submitForgetForm" type="button"><i></i>  بازیابی کلمه
عبور    </button>
                <% } %>
```

اما مشکل اینجاست که این صفحه توی مرورگر موزیلا و کروم به خوبی کار می‌کنه اما توی اینترنت اکسپلورر در حالت Compatibility View وقتی تابع Forget() اجرا میشه ، مقدار itm.User_Name نال میشه.(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. model binder این‌ها رو به هم map نمی‌کنه. ضمنا اگر فرم شما اطلاعاتی رو آپلود نمی‌کنه، encType اون رو حذف کنید.
- دیباگ کنید مرورگر چه اطلاعاتی رو ارسال می‌کنه. از فیدلر هم می‌تونید برای دیباگ 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 برام پیش اومده :
آیا تعریف یک پراپرتی 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 مثلا 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 پراپرتی‌های غیر ضروری ، مشکل امنیتی برطرف میشه ؟
(بدین صورت اگر کاربر شیطونی کنه و مثلا فیلدای IsAdmin رو دستی بسازه و .... ازش استفاده نمیشه و مشکلی پیش نمیاد)
(نهایتا Model.IsValid هم می‌تونیم در اینجا استفاده کنیم)

مسلما موقع ثبت هم مقدار پروپرتی‌های حساس رو خودمون دستی پر می‌کنیم و اصلا 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 ، همش انگلیسی پیام میده!
و متن فارسی منو نادیده می‌گیره ...
یک توضیح بیشتر اگر مرحمت کنین ، ممنون میشم.

با تشکر

نویسنده: ح مراداف
تاریخ: ۱۳۹۲/۱۱/۲۳ ۱:۱۲

با سلام
آیا منظور شما اینه که تاریخ شمسی رو بصورت string توی دیتابیس بریزیم ؟
PersianDate : VARCHAR <<
Date : DATE <<

بدین صورت جهت استخراج تاریخ‌های شمسی و زدن کوئری به یک عدد پروسیجر بزرگ نیاز خواهیم داشت (مثلا بر اساس "/" اسپلیت کنه و بعد هر بخش رو int کنه و ....)
آیا چاره دیگری نداریم ؟ :(

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۲۳ ۱:۲۰

- 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 1/1/92 باید 01/01/92 ذخیره شود (این روش از دوران فاکس‌پرو 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: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۲۳ ۱:۳۲

- روش بهینه، استفاده از یک Context در طول درخواست است. در قسمت 12 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 نوشتن 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 اعتبارسنجی یک قسمت مجزا اختصاص داده شده‌است. جزئیات روش کار خودتان را با آن مقایسه کنید.

نویسنده: علی یگانه مقدم
تاریخ: ۱۳۹۳/۰۹/۲۴ ۱:۴۸

با تشکر از مطلب مفیدتان
شما گفتید در mvc به جای viewstateها از برگشت دادن یک کلاس استفاده می‌کنیم ولی من الان اگر در اکشن نوع httppost فقط بنویسم

```
return view();
```

باز هم داده‌ها داخل فیلدها قرار دارن و هیچ فرقی با برگشت زدن مدل نداره

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۲۴ ۳:۶

- 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 **For** TextBox ، **For** Editor 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 اصلا کاری به model بازگشتی ندارند.
- اگر در این بین نیاز به تغییری در مقدار خاصیتی باشد، نیاز است حتما model بازگشت داده شود؛ ضمنا نکته‌ی ویژه‌ای هم برای به روز رسانی مقدار تغییر کرده وجود دارد (به علت استفاده از ModelState، تغییرات بر روی مدل دریافتی تا پیش از Remove

اعمال نمی‌شوند) :

```
[HttpPost]
public ActionResult Create(User user)
{
   ModelState.Remove("Name"); // به روز رسانی مقدار یک خاصیت در حالت ویژه ارسال به سرور
   user.Name = "new name";
   return View(user);
}
```

- اگر در ViewModel بازگشتی به View، نیاز است مثلا drop down ایی 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 ویژه‌ی 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 سناریوها کار می‌کند توصیه می‌شود.

عنوان: ASP.NET MVC #12
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۲۱ ۰۰:۲۰:۰۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

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 وادار نمی‌کند که مدل را به نحو خاصی طراحی کنید. شما را مجبور نمی‌کند که کلاس‌های مدل را برای نمونه همانند کلاس‌های کنترلرها، از کلاس خاصی به ارث ببرید. یا حتی در مورد نحوه‌ی دسترسی به داده‌ها نیز، نظری ندارد. به عبارتی برنامه نویس است که می‌تواند بر اساس امکانات مهیای در کل اکوسیستم دات نت، در این مورد آزادانه تصمیم گیری 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 نحوه‌ی دربرگیری اطلاعات را در مدل‌ها توضیح می‌دهند. برای دسترسی به آن‌ها پروژه 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 پروژه اینکار را انجام می‌دهد).

یک مثال کاربردی

یک پروژه جدید خالی ASP.NET MVC را آغاز کنید. در پوشه مدل‌های آن، مدل اولیه‌ای را با محتوای 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 محتوای زیر به پروژه 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 Html.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 آن‌را وارد نماید. یا مثلا برچسب AddDate نباید به این شکل صرفا 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 توکار دات نت و ASP.NET MVC استفاده کرد و نهایتا این metadata توسط فریم ورک، مورد استفاده قرار خواهند 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 کرده‌ایم که نوع ورودی فقط قرار است Date باشد و نیازی به قسمت Time آن نداریم.
DataType شامل نوع‌های 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 مجددا 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 اندکی 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 classهای css 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 امکان تبدیل آن مثلا به RadioButton انتخاب مرد یا زن بودن کارمند ثبت شده در سیستم وجود داشت. برای اصلاح یا تغییر این مورد، باز هم می‌توان از متادیتای خواص، جهت تعریف قالبی خاص برای هر کدام از خواص مدل استفاده کرد.
به پوشه Views/Shared مراجعه کرده و یک پوشه جدید به نام EditorTemplates را ایجاد نمائید. بر روی این پوشه 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 آن‌را هم مثلا GenderOptions وارد کنید. همچنین گزینه «Create a strongly typed view» را نیز انتخاب کنید. مقدار Model class را مساوی bool وارد نمائید. فعلا 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 نهایتا به شکل زیر خواهند 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 دراپ داون لیست):

```
[UIHint("Enum")]
```

سایر متدهای کمکی تولید و نمایش خودکار اطلاعات از روی اطلاعات مدل‌های برنامه

متدهای دیگری نیز در رده‌ی Templated helpers قرار می‌گیرند. اگر از متد Html.EditorFor استفاده کنیم، از تمام این اطلاعات متادیتای تعریف شده نیز استفاده خواهد کرد. همانطور که در قسمت قبل (قسمت 11) نیز توضیح داده شد، صفحه استاندارد Add view در VS.NET به همراه یک سری قالب تولید فرم‌های Create و Edit هم هست که دقیقا کد نهایی تولیدی را بر اساس همین متد 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 بی نقص است!

به علاوه، متد Html.DisplayForModel نیز مهیا است. بجای اینکه کار تولید رابط کاربری اطلاعات نمایش جزئیات یک شیء را انجام دهید، اجازه دهید تا متد 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 Viewهای 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 فرمت اطلاعات در حال نمایش را تغییر دهیم، به نحو زیر می‌توان عمل کرد:

```
[DisplayFormat(DataFormatString  = "{0:n}")]
```

مقدار DataFormatString در پشت صحنه در متد string.Format مورد استفاده 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 قرارداهای بکارگرفته شده در حین استفاده از قالب‌های سفارشی، استفاده از نام اشیاء می‌باشد. مثلا در پوشه Views/Shared/DisplayTemplates، اگر یک Partial 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 datetime.cshtml سفارشی را تولید کنید که Model آن تاریخ میلادی دریافتی است و نهایتا کار این Partial 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 Partial view ایی به نام Employee.cshtml در پوشه Views/Shared/DisplayTemplates قرار گیرد، متد 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 دراینجا 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 البته با این تفاوت که Page life cycle و Code behind و سایر موارد مشابه آن در اینجا حذف شده‌اند. همچنین از Partial viewها برای به روز رسانی قسمتی از صفحه حین فراخوانی‌های Ajaxایی نیز استفاده می‌شود. مهم‌ترین کاربرد Partial views علاوه بر استفاده مجدد از کدها، خلوت کردن Viewهای شلوغ است جهت ساده‌تر سازی نگهداری آن‌ها در طول زمان (یک نوع Refactoring فایل‌های View محسوب می‌شوند).
پسوند این فایل‌ها نیز بسته به موتور View مورد استفاده تعیین می‌شود. برای مثال حین استفاده از Razor، پسوند Partial views همان 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 viewها یکی است، مرسوم شده است که نام Partial viewها را با یک underline شروع کنیم تا بتوان بین این دو تمایز قائل شد.

اگر این فایل‌ها را در پوشه Views/Shared تعریف کنیم، در تمام Viewها قابل استفاده خواهند بود. اما اگر مثلا در پوشه Views/Home آن‌هارا قرار دهیم، تنها در Viewهای متعلق به کنترلر Home، قابل بکارگیری می‌باشند.

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 آن‌را 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 جزئی از یک View را نهایتا 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);
}
```

View متناظر با این متد را هم با کلیک راست بر روی متد، انتخاب گزینه Add view و سپس ایجاد یک strongly typed view از نوع کلاس Employee، ایجاد خواهیم کرد.

در ادامه قصد داریم بدنه حلقه زیر را refactor کنیم و آن‌را به یک Parial 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 کلیک راست کرده و گزینه Add|View را انتخاب کنید. در اینجا نام _EmployeeItem را وارد کرده و همچنین گزینه Create as a partial view و create a strongly typed view را نیز انتخاب کنید. نوع مدل هم Employee خواهد بود. به این ترتیب فایل زیر تشکیل خواهد شد:

```
\Views\Employee\_EmployeeItem.cshtml
```

ابتدای نام فایل‌را با underline شروع کرده‌ایم تا بتوان بین Viewها و Partial views تفاوت قائل شد. همچنین این Partial view چون داخل پوشه Employee تعریف شده، فقط در Viewهای کنترلر Employee در دسترس خواهد بود.
در ادامه کل بدنه 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 نوشتن مستقیم در Response است، برخلاف Html.Partial که فقط یک رشته را بر می‌گرداند.

نمایش اطلاعات از کنترلرهای مختلف در یک صفحه

Html.Partial بر اساس اطلاعات مدل ارسالی از یک کنترلر، کار رندر قسمتی از آن‌را در یک View خاص عهده دار خواهد شد. اما اگر بخواهیم مثلا در یک صفحه یک قسمت را به نمایش آخرین اخبار و یک قسمت را به نمایش آخرین وضعیت آب و هوا 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 بر می‌گرداند و دومی اطلاعات را مستقیما 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. مثلا با محتوای زیر (با توجه به اینکه مدل ارسالی از نوع رشته‌ای است):

```
@model string

<ul>
    <li>
        @Model
    </li>
</ul>
```

حین 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 Partial view یک underline قرار دادیم، نیاز خواهد بود تا این نام صریحا ذکر گردد (چون دیگر هم نام متد یا ActionName آن نیست). ویژگی ChildActionOnly سبب می‌شود تا این متد ویژه تنها از طریق فراخوانی Html.Action در دسترس باشد.
برای استفاده از آن هم در Viewایی 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. صرفا یک الحاق 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 مثلا 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 حالتیکه این اطلاعات از قسمت‌های دیگر صفحه ایزوله است (مثلا نمایش چند ویجت مختلف در صفحه)، روش RenderAction ارجحیت دارد.

یک نکته

فراخوانی متدهای RenderAction و RenderPartial در حین کار با Razor باید به شکل فراخوانی یک متد داخل {} باشند:

```
@{ Html.RenderAction("About"); }
And not @Html.RenderAction("About")
```

علت این است که @ به تنهایی به معنای نوشتن در Response است. متد RenderAction هم خروجی ندارد و مستقیما در Response اطلاعات خودش را درج می‌کند. بنابراین این دو با هم همخوانی ندارند و باید به شکل یک متد معمولی با آن رفتار کرد.

اگر حجم اطلاعاتی که قرار است در صفحه درج شود بالا است، متدهای RenderAction و RenderPartial نسبت به Html.Action و 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 ۰۰:۰۱:۵۴

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 روشی 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 ۰۱:۱۳:۳۲

- برای نمایش خودکار تاریخ فارسی می‌تونید از این فایل استفاده کنید (تست شده) : (^)
- 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 می‌تونید 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 : (^) البته این فقط یک ایده برای شروع است که چه فرمت‌هایی می‌تونه وارد بشه و قابل قبول باشه.

نویسنده: Mojtaba
تاریخ: ۱۳۹۱/۰۱/۲۸ ۰۲:۳۹:۰۸

نکته : تفاوت این دو خط منجر به عدم توانایی برنامه برای استفاده ازDisplayTemplates 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 ۱۰:۰۸:۵۳

- اگر قرار است فایل datetime.cshtml فرمت نهایی را اعمال کند، چرا جایی دیگری می‌خواهید 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 ۱۰:۱۴:۵۱

به صورت خلاصه DisplayTemplates فقط به متد Html.DisplayFor اعمال می‌شوند و نه خارج از آن. زمانیکه مستقلا از String.Format استفاده می‌کنید، یعنی خودتان قصد دارید اطلاعات را پردازش و فرمت کنید.

نویسنده: Mojtaba
تاریخ: ۰۲:۴۴:۵۸ ۱۳۹۱/۰۱/۲۹

سلام آقای نصیری خسته نباشید!
اگر بخواهم ازیک jQuery datepicker تاریخ شمسی، مثل آنچه که معرفی کرده اید در EditorTemplates استفاده کنم، چه راه حلی رو پیشنهاد می‌کنید .
ممنونم از لطفتون
ضمنا محبت کنید برای فایل خلاصه‌ی وبلاگ از آخرین تاریخ تهیه به عنوان اسم استفاده کنید تا تاریخ آپدیت آن مشخص شود.

نویسنده: وحید نصیری
تاریخ: ۰۹:۴۶:۲۲ ۱۳۹۱/۰۲/۰۱

می‌تونید داخل templated helper ایی که تعریف می‌کنید یک Html.TextBox اضافه کنید که پارامتر html attributes آن مقدار دهی شده است. چون برای نمونه این datePicker مثلا به TextBox ایی با class خاص اعمال می‌شود. مثلا: "new { @class = "datepicker

نویسنده: Milad Mohseni
تاریخ: ۱۱:۵۷:۰۰ ۱۳۹۱/۰۲/۱۱

با 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. مثلا Html.DropDownListFor یا BeginForm و غیره ...
نهایتاً تمام این ها سمت سرور اجرا شده و نتیجه HTML برمیگرداند. 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 نداشتیم، نتیجهٔ HTML شده را جهت استفاده نهایی استفاده کنیم، یعنی در Vew نهایی که بر روی Host آپلود می شود، کد های HTML تولید شده در مرحله قبل را قرار دهیم.
با این کار دیگر لازم نیست سمت سرور Viewها ساخته شوند و همه چیز آماده است؛ که میتواند باعث افزایش سرعت شود. آیا درست متوجه شدم؟
البته قسمت هایی مثل ActionLink ها که با توجه به موقعیت جاری ساخته میشوند به حالت قبل در View نوشته شوند.

نویسنده: وحید نصیری
تاریخ: ۱۲:۰۷:۴۶ ۱۳۹۱/۰۲/۱۱

- Viewها در ASP.NET MVC پس از اولین بار فراخوانی، کامپایل می‌شوند. البته می‌شود با دستکاری فایل پروژه، آن‌ها را 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 متد کمکی. اما مثلا در مورد رندر کردن یک گرید پویا بر اساس پارامترهای انتخابی 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 اصلا کدی در سمت سرور اجرا نشود. البته در مورد بایدها و نبایدهای آن هم بحث شده در همان مطلب.

نویسنده: Milad Mohseni
تاریخ: ۱۶:۱۷:۳۰ ۱۳۹۱/۰۲/۱۱

بسیار متشکرم.
کاملاً فهمیدم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۱۱ ۱۶:۲۵:۵۱

خواهش می‌کنم. یک مورد دیگر هم هست:
- متدی 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 مثلا یک برچسب یا یک تکست باکس و موارد مشابه، این compile time checking رو از دست خواهید داد.

نویسنده: محمد شهریاری
تاریخ: ۱۳۹۱/۰۳/۳۱ ۱۱:۳۲

هنگام استفاده از PartialView در صورتی که بخواهیم به فرض از یک Enume استفاده کنیم با توجه به توضیحاتی که در بالا امده است و انرا 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 مستقیما در یک View هم می‌تونید 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 DisplayTemplates قرار دادم. اما همچنان DateTimeها رو به فرمت میلادی نمایش میده. 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 DisplayTemplates فقط به متد Html.DisplayFor اعمال می‌شوند و نه خارج از آن. زمانیکه مستقلا از String.Format استفاده می‌کنید، یعنی خودتان قصد دارید اطلاعات را پردازش و فرمت کنید.»

نویسنده: محسن
تاریخ: ۱۳۹۱/۰۵/۰۱ ۱۴:۲۵

سلام آقای نصیری خسته نباشید
یه سوال از خدمتتون داشتم اونم اینکه اگر توی پروژه جایی نیاز باشه که ما اطلاعات یک فیلد رو از دیتابیس بخونیم و توی یک DropDownList نمایش بدیم، 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 به DropDownList بایند میکنیم. حال مطابق با متد Srongly Type View برای متد Create در کنترلر یک View ایجاد میکنیم. همون طور که میدونید Razor به صورت پیش فرض برای فیلد Username یک EditorFor قرار میده. در صورتی که ما میخوایم یک DropDownList به کاربر نشون بدیم که به راحتی بتونه کاربر مورد نظرش رو انتخاب کنه. حالا چجوری میشه این Username که Value این DropDownList هست رو در موقع کلیک بر روی دکمه ذخیره به سمت کنترلر فرستاد؟ در واقع من نمیدونم اصلا میشه Value رو از FormCollection گرفت یا نه ؟ امیدوارم منظورمو خوب بیان کرده باشم
و یه سوال دیگه اینکه در موقع ویرایش چجوری میشه Vaule 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 پیشنیاز 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 یک رکورد مهم است نه تمام خواص و فیلدهای 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 SelectListItemها 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 نگاشت‌ها هم مباحث 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 Model چیه؟

بعدش اینکه برای فهم قطعه کد زیر باید با چه مفاهیمی آشنا باشیم (پر کردن DropDown)

```
@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 : (^ و ^)

- پیشنیازها:

Enum

LINQ

نویسنده: امیر
تاریخ: ۱۳۹۱/۰۷/۲۲ ۱۲:۶

سلام

شرمنده من همچنان کاملا متوجه نشدم فرق model با Model چیه؟

چرا تو view بالای صفحه یه @mode از یکی مدلهای خودمون وهله سازی میکنیم اما پایینتر وقتی خواستیم از اون مدل وهله سازی شده استفاده کنیم مینویسم 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 صریحا ذکر شود.

زمانیکه از Model با M بزرگ استفاده می‌شود، یعنی با یک وهله 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 رو از نوع bool قرار میدم error مید

```
The model item passed into the dictionary is null
```

وقتی create فراخوانی میشه و بعدش partial رو فراخوانی میکنه . مدل partial view نال هست
اما وقتی که partial view رو بدون مدل set کردم مشکلی پیش نیومد و به فیلدم 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:
RadioButtonList در ASP.NET MVC
CheckBoxList در ASP.NET MVC

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 پیشنیازهای 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 Dll دیگر است کار نمی‌کند.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۳۰ ۱۴:۱۴

مشکلی 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

ممنون از پاسختون و همچنین پروژه ای که Attach کردید.

پس از مقایسه و پیگیری متوجه شدم اگر GenderOptions در فولدر Shared\EditorTemplates باشد کار می‌کند اما من این UiHint و صرفا برای یک ویو خاص می‌خوام. از این رو اگز این UidHint و در هر مسیری به غیر از مسیر فوق قرار بدم شناسایی نمیشه. این مسئله مخصوصا به این شکل است و یا من درجایی اشتباه کردم.

یک سوال دیگه که برام پیش اومده این است که ویو من اتوماتیک و با استفاده از متد @Hml.Editorformodels

ساخته میشه اما در متد Post مدلی به ویو پاس ندادم و صرفا return view() زدم و در متد 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 میشه ! و این ساخته شدن مشخص نیست به چه شکله چرا که متد سازنده کلاس (Constructor) را هم صدا نمی‌زند!

باز هم ممنون از پاسختون.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۳۱ ۱۶:۴۷

- در متن توضیح دادم: « 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 Viewهای کنترلر *home* قابل استفاده خواهند بود.»
در متن عنوان شده DisplayTemplates، شما این رو مثلا به EditorTemplates تغییر بدید. اصول یکی است.
- مراجعه کنید به سورس ASP.NET MVC و قسمت‌های مرتبط رو مطالعه کنید؛ جهت آشنایی بیشتر با سازوکار درونی آن‌ها.

نویسنده: ahmadb7
تاریخ: ۱۳۹۲/۰۳/۱۱ ۲۰:۱۸

```
public enum MyGrade { A = 20, B =15, C =10, }
        [UIHint("Enum")]
        public MyGrade Grade { set; get; }
```

در فولدر EditorTemplates ،
partial view مانند مثال شما ایجادکردم اما زمانی که یک view ایجاد میکنم
Dropdownlist تولید نمی‌شود

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۱۱ ۲۲:۱۶

سورس کامل این سری از اینجا و یا سورس قسمت 12 از اینجا قابل دریافت است. نام فایل‌ها، مسیرها و محتوای آن‌ها رو با کار خودتون مقایسه کنید.

نویسنده: لیلا
تاریخ: ۱۳۹۲/۰۴/۲۹ ۹:۳۹

با سلام و خسته نباشید
فایل این سری(12) حاوی مثال Dropdownlist نمی‌باشد و متاسفانه 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

نویسنده: سامان
تاریخ: ۱۳۹۲/۰۴/۲۹ ۱۴:۲۳

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 داره که در runtime به مدل مورد نظر کلاس metadata اضافه شود؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۲۹ ۱۸:۵۹

- 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 نهایتا از طریق Reflection قابل دسترسی خواهند بود. تغییر آن‌ها یا افزودن آن‌ها عموما از طریق دستکاری کدهای IL میسر است یا از روش‌های IL Code weaving مباحث AOP یا روش‌هایی مانند Reflection.Emit و همانند آن.
- 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.

نویسنده: حمیدی
تاریخ: ۱۳۹۲/۰۶/۰۱ ۱:۱۱

سلام؛ من می‌خوام به layout توی پروژه چندتا model 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.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۶/۰۱ ۸:۳۲

- 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 مثلا می‌تونید توسط ViewBag، عنوان صفحه را که در layout تعریف شده، مقدار دهی کنید.
اگر می‌خواهید Strongly typed کار کنید، روش Html.RenderAction یک راه حل است و روش 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 بری کنند. مثلا:

```
public class HomeViewModel : BaseViewModel
{
   public int Data1 { set; get;}
   // ...
}
```

در این حالت و با رعایت این شرط، می‌تونید 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 Viewها ارسال می‌شوند، به صورت strongly typed قابل دسترسی خواهند بود.

نویسنده: حمیدی
تاریخ: ۱۳۹۲/۰۶/۰۱ ۹:۲۶

این که گفتید کاملا انجام شد و ممنون؛ اما من میخوام از foreach استفاده کنم. چطور یه لیست پاس بدم که مثلا 10 مطلب آخر و به صورت desc نشون بده..

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۶/۰۱ ۹:۴۳

- 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 View (تمام Viewهای سایت)، تولید شود. بنابراین در BaseViewModelایی 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; } // خاصیت عمومی که قرار است در فایل مستر قابل دسترسی باشد
}

// در اکشن متد
 return View(model: new HomeViewModel { Posts = .... });

// در اینجا ویومدل ارسالی از پایه مدل‌ها مشتق می‌شود
public class HomeViewModel : BaseViewModel
```

- و اگر نمی‌خواهید به ازای هر Viewایی که در سایت تولید می‌شود یکبار این لیست را مقدار دهی کنید، بهتر است از روش Html.RenderAction استفاده کنید (بحث «نمایش اطلاعات از کنترلرهای مختلف در یک صفحه» در مطلب فوق). مباحث Caching را هم برای لیست‌های عمومی فراموش نکنید.

نویسنده: رضا منصوری
تاریخ: ۱۳۹۲/۰۸/۱۷ ۸:۵۲

با تشکر از مطلب خوبتون
امکان استفاده از RenderAction در Shared/_Layout وجود نداره ؟ وقتی اینکارو انجام میدم با ارور StackOverflowException برخورد میکنم تکه کد اجرای RenderAction انقدر تکرار میشه تا به این ارور برخورد مبکنه!

میخوام لیست گروهبندی مطالبمو تو _Layout.cshtml (نمایش در تمام صفحات مثل MasterPage در WebForm) نشون بدم چه راهی رو پیشنهاد میدید ؟ خیلی ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۸/۱۷ ۹:۳۵

- مشکلی از این لحاظ نیست که نشود از RenderAction در فایل layout استفاده کرد.
- منطق تهیه گروه بندی مطالب شما احتمالا یک حلقه بی‌نهایت دارد یا یک الگوریتم بازگشتی بی پایان است.
- ممکن است در این حالت از return PartialView استفاده نکردید و return View بوده . در این حالت View 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 رندر شدن در فایل layout دوباره سبب ارجاع بی‌نهایتی به فایل Layout نشود.

نویسنده: رضا منصوری
تاریخ: ۱۳۹۲/۰۸/۱۷ ۱۵:۲۲

حدس شما درست بود از return PartialView استفاده نکرده بودم بازم ممنون

نویسنده: م رمضانی
تاریخ: ۱۳۹۲/۱۱/۱۱ ۸:۴۶

سلام؛ اگر بخواهیم textboxئی که در view برای فیلدی با مورد مصرفی currency ولی data type: decimal (در لایه مدل) 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 textboxئی 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)؟
متشکرم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۱ ۹:۴۹

- دومین DisplayFormat ایی که در متن هست، دقیقا 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 ViewModel 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: ۱۳۹۲/۱۱/۱۱ ۱۱:۱۲

برای پیاده سازی سه رقم جداکننده به این شیوه عمل کردم:

در Model

```
[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 متاسفانه جوابی نگرفتم! (برای طراحی view از bootstrap استفاده میکنم)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۱ ۱۲:۱۳

- DisplayFormat برای کارهای نمایشی سمت سرور است. مثلا در یک حلقه Razor سمت سرور، لیستی را نمایش می‌دهید؛ متد Html.DisplayFor از آن استفاده خواهد کرد.

- اگر قرار است در حین ورود اطلاعات کاربر، به صورت خودکار سه رقم جداکننده 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: ۱۳۹۲/۱۱/۱۸ ۱۹:۳

سلام؛ جوری که من متوجه شدم در Razor وقتی تعدادی ماژول داریم (مثل آخرین اخبار و دسته بندی مطالب و ...) باید Controller و view مخصوص به خودش رو ایجاد کنیم و توی Layout برنامه با متدهای Html.RenderAction و Html.Action اون هارو توی Sectionهای دلخواه فراخوانی کنیم تا توی همه پیج‌ها نمایش داده بشن درسته؟ ممنون.

نویسنده: ح مراداف
تاریخ: ۱۳۹۲/۱۱/۲۳ ۳:۱

سلام،
بله، طبق علم بنده، توی هر View فقط با یک مدل میشه کار کرد (یک indstance 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 ویو 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 کدوم یک دیتای مجزان مسلما 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 پیج (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 Partial View داریم..... همین!
به جای درگ کردن یوزرکنترل توی مستر پیج هم RenderAction و Action داریم...
{ کلا MVC اومده 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 صفحاتتو ببینی که چقدر تمیز و خوشگل ! شده ، ازش خوشت میاد :)))

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۱/۲۳ ۹:۳۷

در یک View امکان استفاده از چند مدل هم هست. اگر بخواهید dynamic کار کنید می‌شود از ViewBag استفاده کرد اگر بخواهید strongly typed کار کنید می‌شود از ViewModelها استفاده کرد: « نحوه استفاده از ViewModel در ASP.NET MVC »

نویسنده: م رمضانی
تاریخ: ۱۳۹۲/۱۱/۲۵ ۱۲:۳

ممنون؛ به سایتی که معرفی کردید رفتم و هر دو فایل jquery.price_format.2.0.js و jquery.price_format.2.0.min.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: ۱۳۹۲/۱۱/۲۵ ۱۲:۴۹

- نحوه استفاده از افزونه Firebug برای دیباگ برنامه‌های ASP.NET مبتنی 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'
```

عنوان: ASP.NET MVC #13
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۲۲ ۲۲:۳۵:۰۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

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 متادیتا، 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 نمائید:

```
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 Redirect("/");
            }
            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 انتخاب کنید. همچنین قالب Scaffolding را نیز بر روی Create قرار دهید.

توضیحات تکمیلی

همانطور که در مدل برنامه ملاحظه می‌نمائید، به کمک یک سری متادیتا یا اصطلاحا 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، حداکثر 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، عموما 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 مشخص می‌کند که آیا کارmodel 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 مجددا 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. بلافاصله 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 صرفا 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 مجددا 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.unobtrusive متعلق به تیم ASP.NET MVC است و کار آن وفق دادن سیستم موجود، با jQuery 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.

Unobtrusive همچنین در اینجا به معنای مجزا سازی کدهای جاوا اسکریپتی، از سورس HTML صفحه و استفاده از ویژگی‌های data-* 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 را false کرده و بار دیگر برنامه را اجرا کنید. در این حالت کلیه کدهای اعتبار سنجی، به داخل سورس View رندر شده، تزریق می‌شوند و مجزا از آن نخواهند بود.

نحوه‌ی تعریف این اسکریپت‌ها نیز جالب توجه است. متد Url.Content، یک متد سمت سرور می‌باشد که در زمان اجرای برنامه، مسیر نسبی وارد شده را بر اساس ساختار سایت اصلاح می‌کند. حرف ~ بکارگرفته شده، در ASP.NET به معنای ریشه سایت است. بنابراین مسیر نسبی تعریف شده از ریشه سایت شروع و تفسیر می‌شود.

اگر از این متد استفاده نکنیم، مجبور خواهیم شد که مسیرهای نسبی را به شکل زیر تعریف کنیم:

```
<script src="../../Scripts/customvaildation.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 متد 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 متد 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).
متد Html.ValidationMessageFor، با توجه به متادیتای یک خاصیت و همچنین استثناهای صادر شده حین model binding خطایی 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 متد IsValid آن‌را تحریف کنیم. اگر این متد 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 مجددا برنامه را اجرا نمائید. 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 متد 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 new ValidationResult، یک پیغام را به همراه فیلدهایی که باید این پیغام را نمایش دهند، بازگردانیم.
به این نوع مدل‌ها، self validating models هم گفته می‌شود.

یک نکته:

از MVC3 به بعد، حین کار با ValidationAttribute، امکان تحریف متد IsValid به همراه پارامتری از نوع ValidationContext نیز وجود دارد. به این ترتیب می‌توان به اطلاعات سایر خواص نیز دست یافت. البته در این حالت نیاز به استفاده از Reflection خواهد بود و پیاده سازی 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 ASP.NET 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، 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 customvaildation.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/customvaildation.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 خواهند 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 ویژگی 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 صرفا وظیفه ارائه متادیتاهای 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 مثلا 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 Ajaxایی خودکار را به سرور ارسال خواهد کرد و نتیجه نهایی را به کاربر نمایش می‌دهد؛ کارهایی که به سادگی توسط کدهای جاوا اسکریپتی 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 خواهند شد. همچنین اگر نیاز بود فیلدهای دیگری نیز به این متد کنترلر ارسال شوند، می‌توان آن‌ها را توسط خاصیت AdditionalFields، مشخص کرد.

سپس در کدهای کنترلر مشخص شده، متدی با پارامترهای خاصیت مورد نظر و فیلدهای اضافی دیگر، تعریف می‌شود. در اینجا فرصت 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 درخواست‌های Ajaxایی جلوگیری کرده‌ایم. 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 برمتادیتا در خارج از ASP.Net MVC

مباحثی را که در این قسمت ملاحظه نمودید، منحصر به ASP.NET MVC نیستند. برای نمونه توسط متد الحاقی زیر نیز می‌توان 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، به برنامه اضافه کنیم و تمام عملیات هم دستی است و فریم ورک ویژه‌ای هم وجود 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 مانند onfocusout، onkeyup، onclick و غیره مقدار دهی شوند: (^) و (^)
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 WebGird هم Edit 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 تعریف شده در صفحه به صورت جداگانه انجام داد. به این ترتیب بجای اینکه کل گرید را در یک فرم تعریف 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 اعتبارسنجی تاریخ شمسی در mvc و در بالای view model میتونید 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 گفتین من توسط MetaData برای فیلد Username 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 متد اعتبار سنجی جدید که می‌تونه کوئری صحیح بر اساس کاربر جاری لاگین 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 بدید.

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 Membership و ProfileBase خود دات نت توی پایگاه داده میریزیم. الان منظور شما اینه که من باید برای ویرایش یک کلاس CustomProfile دیگه با همون مشخصات بسازم و متد اعتبار سنجی برای Username اون رو یه متد دیگه قرار بدم ؟ ببینید 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 اومدم و این کلاس را به عنوان ViewModel خودم قرار دادم. و وقتی فرم Create و یا Edit رو Submit میکنم متد اعتبار سنجی من وارد عمل میشه

اینم کدViewها:

```
@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)
```

```
                            @Html.ValidationMessageFor(model => model.NationalCode)
                        </div>
                    </td>
                </tr>
                <tr>
                    <td>
                        <div>
                            @Html.LabelFor(model => model.Email, "ایمیل")
                        </div>
                    </td>
                    <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>
                    <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>
                    <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>
                    <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 ViewModel دیگه مثلا CustomProfileEdit استفاده کنم و اونجا متد

اعتبار سنجی دیگه ای تعریف کنم. منظور شما همین بود ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۲۰ ۱۲:۵

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: محمدرضا
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۴:۴۹

میخواستم بدون آیا استفاده از Data Annotationها 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 Annotationها رو تعریف کرد و بعد با Auto Mapper به هم Map کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۷:۳۴

بله. مشکلی نداره. در EF Code first از Data Annotationها 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 کردید، 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.

ViewModel هم باید Data Annotation مختص به خودش را داشته باشد. حداقل روی اعتبار سنجی سمت کلاینت می‌تونه تاثیرگذار باشه چون به صورت خودکار توسط MVC اعمال می‌شود.

نویسنده: رضا
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۸:۳۱

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 Annotationها و Map کردن به کلاسی که برای EF درست کردیم، یا روی همون کلاس‌های EF Code First؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۹ ۱۸:۳۹

امکانات Data Annotationها کار تیم EF یا به عبارتی طراحان اصلی دات نت است. بنابراین کمی زیاده روی است که عنوان کنیم غیراصولی هستند. شما در NHibernate چندین و چند روش تعریف نگاشت‌ها و روابط را دارید. از فایل‌های XML تا Data Annotationها تا Fluent NH تا Mapping by code اخیر آن و غیره. 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 Data Annotationها و Fluent

API استفاده می‌کنم.
ViewModel فقط بحث مدیریت صحیح ارتباط با کلاینت است (نمایش اطلاعات View) و برعکس (دریافت اطلاعات از کاربر).
بنابراین اگر از ViewModel استفاده می‌کنید، نیاز است از Data Annotationها استفاده کنید تا اعتبار سنجی سمت کاربر (که به صورت خودکار توسط MVC اعمال و مدیریت می‌شود) کار کند.

نویسنده: طاهریان
تاریخ: ۱۳۹۱/۰۷/۱۱ ۲۲:۲۴

سلام
برداشت من از صحبت شما این هست که زمانی که از ViewModel استفاده می‌کنیم DataAnotation رو باید داخل ViewModel تعریف کنیم.
با این کار اصل DRP زیر سوال نمیره؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۲۰ ۰:۲۶

آ یا استفاده از ویژگی‌ها (Attributes یا Data Annotations) خلاف SRP یا Single Responsibility Principle کلاس‌ها است؟
پاسخ: خیر.
توضیحات:
این سؤال از اینجا ناشی می‌شود که چون برای مثال با استفاده از ویژگی StringLength می‌توان طول یک خاصیت رشته‌ای را مشخص کرد و اگر طول وارد شده توسط کاربر بیش از مقدار تعیین شده باشد، استثنایی صادر می‌شود یا مکانیزم‌های اعتبار سنجی سمت کاربر فعال خواهند شد، بنابراین کلاس تعریف 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 متادیتای تعیین شده، دارد. برای مثال این ویژگی‌ها را در یک برنامه ساده کنسول تعریف کنید. هیچ اتفاقی رخ نخواهد داد. زیرا در این حالت متادیتای تعریف 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: ۱۳۹۱/۱۱/۰۵ ۱۱:۲۷

با سلام.
در حالت زیر در هنگام submit همواره صفحه رفرش می‌شود و بعد از رفرش صفحه خطاها را نشان می‌دهد و اعتبارسنجی 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

احتمالا اسکریپت‌های شما درست load نشده.

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 لاگین 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 متشکرم.

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 برطرف شد. بسیار متشکرم 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 mvc 4 استفاده می‌کنم.

با تشکر.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۱۵ ۱۶:۴۵

منتقل شده به اسمبلی System.Web.Webpages نگارش 2 آن.

نویسنده: سعید یزدانی
تاریخ: ۱۳۹۱/۱۱/۱۸ ۲۳:۱۶

چرا ما در پیاده سازی متد 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 البته. به دلیل امضای متد 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.readyنباشد 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");
```

value "0001/01/01 12:00:00 ق.ظ"

کدهای کامل این سری همانطور که در مطالب قبلی هم عنوان شده از آدرس ذیل قابل دریافت است:

MVC_Samples

مراجعه کنید به پوشه MVC-13 آن که حاوی کدهای قسمت 13 است.

نویسنده: Leila_gh
تاریخ: ۱۳۹۲/۰۵/۰۲ ۹:۱۹

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 می‌کند.

ولی بعد از اینکه دوباره مثال را در mvc4 vs2012 باز نویسی کردم کار نکرد (دقیقا مانند مثال code project) پس از تغییر متداز (- split) به (/) split و تغییر

```
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 می‌کرد، 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: ۱۳۹۲/۰۵/۰۲ ۱۱:۲۳

کتابخانه‌های جی‌کوئری و کلیه مشتقات آن تقریبا ماهی یکبار شخم زده می‌شوند. اگر به کنسول پاور شل نیوگت در VS.NET مراجعه و دستور update-package را صادر کنید، متوجه حجیم عظیمی از تغییرات خواهید شد. تمام وابستگی‌ها هم اخیرا تغییر کردند. بنابراین اگر از یک مجموعه ناهماهنگ استفاده کنید، درسته ... ممکنه چیزی کار نکند.
مثلا jquery.validate.unobtrusive با jquery.validate و jquery «باید» هماهنگ باشند و اگر نیستند دستور update-package را فراموش نکنید. ضمنا پس از به روز رسانی، «باید» ارجاعات به فایل‌های جدید را در viewهای برنامه درست کنید. مثلا 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.

ضمنا من مجددا 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 DomainClasses کلاس CustomValidator 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 ازMetadata 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)
```

عنوان: ASP.NET MVC #14
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۲۵ ۰۰:۱۷:۰۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

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 کرده باشید. این نوع مفاهیم برای برنامه نویس‌های ASP.NET Web forms آشنا است (و اگر با Web forms view engine در ASP.NET MVC کار کنید، دقیقا یکی است؛ البته با این تفاوت که فایل code behind آن‌ها حذف شده است). به همین جهت در این قسمت برای تکمیل بحث، مروری خواهیم داشت بر نحوه‌ی معرفی جدید آن‌ها توسط Razor.
در یک پروژه جدید ASP.NET MVC و در پوشه 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 Viewایی که تا این قسمت به پروژه‌های مختلف اضافه کردیم، تگ‌هایی مانند body، title و امثال آن وجود نداشتند. در ASP.NET مرسوم است کلیه اطلاعات تکراری صفحات مختلف سایت را مانند تگ‌های یاد شده به همراه منویی 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 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 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 را مشاهده می‌کنید. در اینجا توسط متد RenderBody، محتوای رندر شده یک View درخواستی، به داخل تگ body تزریق خواهد شد.
تا اینجا در تمام مثال‌های قبلی این سری، فایل layout در Viewهای اضافه شده معرفی نشد. اما اگر برنامه را اجرا کنیم باز هم به نظر می‌رسد که فایل layout 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. مثلا یک layout ویژه برای مرورگرهای موبایل‌ها و layout ایی 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 درنظر داشت که می‌توان فایل ViewStart را در زیر پوشه‌های پوشه اصلی View نیز قرار داد. مثلا اگر فایل ViewStart ایی 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 مثلا 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 ایی که در فایل View تعریف شده، در فایل layout جهت رندر مقدار تگ title استفاده می‌شود (انتقال یک متغیر از View به یک فایل master page؛ کلاس layout، مدل View ایی 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 حتما باید از متد سمت سرور Url.Content کمک گرفت. در نگارش چهارم ASP.NET MVC، این محدودیت برطرف شده و دقیقا همانند متغیر Layout ایی که در بالا مشاهده می‌کنید، می‌توان بدون نیاز به متد Url.Content، مستقیما 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 متد RenderSection، آن قسمت نمایش داده خواهد شد. در غیراینصورت، محتوای پیش فرضی را در صفحه قرار می‌دهد. البته اگر از متد RenderSection با آرگومان required: false استفاده شود، درصورتیکه View جاری حاوی قسمتی به نام مثلا menu نباشد، تنها چیزی نمایش داده نخواهد شد. اگر این آرگومان را حذف کنیم، یک استثنای عدم یافت شدن ناحیه 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 تعریف شده در master page مورد نظر، یک سطر زیر را اضافه کرد:

```
@RenderSection("JavaScript", required: false)
```

به این ترتیب، اگر view ایی حاوی تعریف قسمت JavaScript نبود، به صورت خودکار شامل تعاریف الحاق اسکریپت‌های یاد شده نیز نخواهد گردید. در نتیجه دارای حجمی کمتر و سرعت بارگذاری بالاتری نیز خواهد بود.

مدیریت بهتر فایل‌ها و پوشه‌های 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 صدها 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 آن‌را 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. مثلا 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 Views، Model و Controllers 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 نحوه‌ی مسیریابی آن‌ها نیز آشنا شویم. برای این منظور فایل Global.asax.cs قرار گرفته در ریشه اصلی برنامه را باز کنید. در متد Application_Start، متدی 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 AreaRegistration به صورت خودکار یافت شده و پردازش خواهند شد. این کلاس‌ها هم به صورت پیش فرض به نام SomeNameAreaRegistration.cs در ریشه اصلی هر Area توسط VS.NET تولید می‌شوند. برای نمونه فایل ReportingAreaRegistration.cs تولید شده، حاوی 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 منحصربفرد 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 قرارگیرند. سایر مباحث آن هم مانند قبل است. برای مثال در اینجا نام اکشن متد پیش فرض، 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 مثلا 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 پوشه‌های 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 متد پیش فرض Index هر کدام هم یک View جدید را با کلیک راست بر روی نام متد و انتخاب گزینه Add view، اضافه کنید. اکنون برنامه را به همین نحو اجرا نمائید. 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 آن‌را جستجو خواهد کرد). به همین جهت فایل Global.asax.cs را گشوده و متد RegisterRoutes آن‌را مثلا به نحو زیر اصلاح نمائید:

```
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 مجددا برنامه را اجرا کنیم، بدون مشکل View متناظر با متد Index کنترلر Home نمایش داده خواهد شد.
البته این مشکل با نواحی ویژه و غیر اصلی سایت وجود ندارد؛ چون جستجوی پیش فرض کنترلرها بر اساس ناحیه است.

در ادامه مسیر http://localhost/Admin/Home را نیز در مرورگر وارد کنید. سپس بر روی صفحه در مروگر کلیک راست کرده و سورس صفحه را بررسی کنید. مشاهده خواهید کرد که master page یا فایل layout ایی به آن اعمال نشده است. علت را هم در ابتدای بحث Areas مطالعه کردید. فایل Views_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 ناحیه جاری اشاره کند باید نام Area را صریحا ذکر کرد:

```
@Html.ActionLink(linkText: "Home", actionName: "Index", controllerName: "Home",
                 routeValues: new { Area = "Admin" } , htmlAttributes: null)
```

همین نکته را باید حین کار با متد 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 نباشید بابت این دوره ی 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 می شویم:

The following sections have been defined but have not been rendered for the layout page
"~/Views/Shared/_Layout.cshtml": "Menu"."ضمن اینکه دستور در داخل بلاک است!(؟)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۲۶ ۱۶:۰۶:۲۶

بله. درسته. علت هم این است که خروجی RenderSection نهایتا 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 سایت های خارجی دیدم ولی اکثرا با جواب نمی داد یا فقط رو mvc 2 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 مستقیما 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 می‌تونید اینکار رو انجام بدید. یا اینکه باید کد نویسی کنید: (^)

نویسنده: Foad abdollahi
تاریخ: ۱۳۹۱/۰۲/۱۰ ۲۲:۲۰:۵۳

ممنون که جواب دادید
ولی من این 2 تا پست را دیدم که ادعا دارن که با route می شه این کار رو 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 نتونستم درست پیاده سازی کنم ولی از بعضی کامنت ها اینطور معلوم بود که جواب گرفتن

شما نظرتون چیه؟

من در مورد url rewrite تو asp form 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: (^)

نویسنده: میثم
تاریخ: ۱۳۹۱/۰۴/۲۶ ۱۰:۱۹

سلام و خسته نباشید
من فایل _ViewStart.cshtml رو به ریشه منتقل کردم اما خطای زیر رو دریافت کردم ممنون 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 masterpageهای تو در تو شبیه به ASP.NET Web Forms وجود دارد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۲۷ ۹:۴۵

- 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: ۱۳۹۱/۱۲/۲۶ ۹:۵۴

سلام
من نیاز دارم که به صفحه مستر خودم یک مدل پاس بدهم و قسمتهایی از مسترپیج با آن مدل پر شود
چطور می‌توانم این کار را انجام دهم؟(یعنی می‌خواهم به ازای همه اکشن متد هایی که کاربر فراخوانی می‌کند یک اکشن متد دیگه هم که مربوط به صفحه _layout هست هم فراخوانی شده و مقداری را به _layout.cshtml ارسال کند)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۲۶ ۹:۵۹

از Html.RenderAction استفاده کنید.

نویسنده: debuger
تاریخ: ۱۳۹۲/۰۳/۰۷ ۹:۲۹

با سلام؛ آیا این امکان وجود دارد که هر Area به صورت یک Assembly مجزا تبدیل گردد ؟
آیا Area روشی برای ماژولار بودن برنامه‌های MVC است ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۰۷ ۹:۴۰

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: ۱۳۹۲/۰۳/۰۷ ۱۱:۴۷

البته شما می‌تونین فایل‌های View را به صورت یک Resource در اسمبلی قرار بدین اما باید خودتون یک انجین سفارشی برای بازیابی آنها طراحی کنید.

نویسنده: محمد میرزایی

تاریخ: ۱۳۹۲/۰۵/۲۹ ۱۶:۲۷

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 = "Adminstration" } ,
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 می‌گرده

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۵/۲۹ ۱۷:۳۲

- 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 Area 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: ۱۳۹۲/۰۸/۲۰ ۱۳:۲۴

با تشکر از مطلب عالیتون در استفاده از @RenderBody در _Layout(مستر پیجم) وقتی @RenderBody را در جای مد نظر خودم استفاده میکنم لینک‌های Post در صفحات کار نمیکنه مثلا در کنترلر لوگین وقتی روی دکمه‌ی Submit کلیک میکنم به Action مورد نظر نمیرود و چنین Url تشکیل می‌شود

localhost:9764/Account/Login?__RequestVerificationToken=nBQV3rE8qXQ6Z_Qchu-Ns6gdj5xo3kAxmRgACFZMmQGcKQHgASI

ولی وقتی @RenderBody رو ابتدای _Layout(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 نمیکنه (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 متد و کنترلر خاصی اشاره می‌کنند. فرق دارد با وب فرم‌های تک فرمه.
- در MVC نباید یک فرم را به صورت کلی در فایل layout قرار داد. باید در جای مناسب در View مرتبط قرار گیرد آن‌هم با ذکر Html.BeginForm تا action فرم به صورت خودکار به اکشن متد متناظر در کنترلری خاص هدایت شود. آن هم 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 حتما به اکشن متد مدنظر شما هدایت شود. در این حالت محتویات آن به آدرس جاری صفحه ارسال می‌شوند (یعنی اکشن متد پیش فرض تعریف شده در مسیریابی سایت).
- زمانی اجزای فرم در Url به صورت کوئری استرینگ نمایان می‌شوند که متد ارسال اطلاعات Get باشد و نه FormMethod.Post (اگر نحوه ارسال اطلاعات صریحا 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 مثلا 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 می‌باشد:

```
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 تاکید می‌کنم که روی لوکال بدون هیچ مشکلی اجرا می‌شود. اما روی سرور که میرود انگار action فرم مقدار دهی نشده.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۰۶ ۱۱:۳۲

در نگارش‌های قبلی MVC (یعنی هاست شما آنچنان به روز نیست) اگر در Route تعریف شده دو پارامتر اختیاری کنار هم قرار می‌گرفتند، سبب می‌شدند تا URL تولید شده برای آن‌ها خالی باشد (^). بنابراین در Admin_default11 تعریف 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 بطور کلی ساید بار ها...

وقتی که ما متد RenderBody() رو در MasterPage تعریف میکنیم برای تعریف سایدبار باید در هر View اونو تعریف کنیم(همون اخرین اخبار) ؟ یا اینکه در همون فایل Layout تعریف میشن؟

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 تعریف و رندر هستند.

در اینجا برای کش کردن و کاهش بار سیستم می‌توان یک ChildAction خاص را طراحی کرد که Partial View آن متشکل از چند

Html.RenderPartial باشد. زمانیکه OutputCache روی آن قرار داده می‌شود، تمام زیر مجموعه‌ها با هم و یکباره کش خواهند شد.

برای مثال اگر قسمت سمت راست صفحه از 5 ویجت تشکیل می‌شود، نیازی نیست 5 بار از Html.RenderAction در فایل Layout استفاده کنید. یک اکشن متد کلی طراحی کنید که توسط ViewModel ایی مشخص، دیتای View متشکل از چند Partial View خودش را که از چند Html.RenderPartial استفاده می‌کند، تامین کند. بعد در فایل Layout فقط همین تک اکشن متد OutputCache دار را توسط Html.RenderAction رندر کنید.

نویسنده: م سلیمانی
تاریخ: ۱۳۹۳/۱۱/۰۸ ۱:۳۳

با سلام

چطور میشه یه Area رو به عنوان یه پروژه جدا تو پترن‌های چند لایه ایجاد کرد.

مثلا تو NopCommerce این کار انجام شده. Admin Area به عنوان یه پروژه جدا تعریف شده!

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۱/۰۸ ۱:۴۴

« توزیع پروژه‌های ASP.NET MVC بدون ارائه فایل‌های View آن »

عنوان: ASP.NET MVC #15
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۲۶ ۲۳:۱۹:۰۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

فیلترها در ASP.NET MVC

پایه قسمت‌های بعدی مانند مباحث امنیت، اعتبار سنجی کاربران، caching و غیره، مبحثی است به نام فیلترها در ASP.NET MVC. تابحال با سه فیلتر به نام‌های NonAction، ActionName و AcceptVerbs آشنا شده‌ایم. به این‌ها Action selector filters هم گفته می‌شود. زمانیکه قرار است یک درخواست رسیده به متدی در یک کنترلر خاص نگاشت شود، فریم ورک ابتدا به متادیتای اعمالی به متدها توجه کرده و بر این اساس درخواست را به متدی صحیح هدایت خواهد کرد. ActionName، نام پیش فرض یک متد را بازنویسی می‌کند و توسط AcceptVerbs اجرای یک متد، به افعالی مانند DELETE، GET، POST 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 دارند 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 متد، اجرا شوند. 4 نوع فیلتر در ASP.NET MVC وجود دارند:

الف) IAuthorizationFilter

این نوع فیلترها پیش از اجرای هر متد یا فیلتر دیگری در کنترلر جاری اجرا شده و امکان لغو اجرای آن‌را فراهم می‌کنند. پیاده سازی پیش‌فرض آن توسط کلاس AuthorizeAttribute در فریم ورک وجود دارد.
بدیهی است این نوع اعمال را مستقیما 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 آن‌را در طول زمان مشکل خواهد ساخت.

ب) IActionFilter

ActionFilterها پیش (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 پیاده سازی شده است.

ج) IResultFilter

ResultFilter بسیار شبیه به ActionFilter است با این تفاوت که تنها پیش از (OnResultExecuting) بازگرداندن نتیجه متد و همچنین پس از (OnResultExecuted) اجرای متد، فراخوانی می‌گردد. کلاس ActionFilterAttribute موجود در فریم ورک، پیاده سازی پیش فرضی از آن‌را ارائه می‌دهد.

د) IExceptionFilter

ExceptionFilterها پس از اجرای تمامی فیلترهای دیگر، همواره اجرا خواهند شد؛ صرفنظر از اینکه آیا در این بین استثنایی رخ داده است یا خیر. بنابراین یکی از کاربردهای آن‌ها می‌تواند ثبت وقایع مرتبط با استثناهای رخداده باشد. پیاده سازی پیش فرض آن توسط کلاس HandleErrorAttribute 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 متدی و همچنین 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 نمائید:

```
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 متد، به برنامه اضافه نمائید. 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 متد درخواستی جاری اجرا 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 مثلا دو نوع ActionFilter متفاوت به یک متد 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 AuthorizationFilterB اجرا می‌گردد و سپس AuthorizationFilterA.
علاوه بر این‌ها محدوده اجرای فیلترها نیز بر بر این حق تقدم اجرایی تاثیر گذار هستند. برای مثال در پشت صحنه زمانیکه قرار است یک فیلتر جدید اجرا شود، وهله 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 می‌نمائید:

```
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.

تعریف 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 دقیقا 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 امثال آن که پیشتر ذکر شد، در یک کنترلر نیز وجود دارد.
کلاس کنترلر دارای محدوده اجرایی First و Order ایی 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، ValidateInpute و غیره است که توضیحات بیشتر آن‌ها به قسمت‌های بعد موکول می‌گردد.

نظرات خوانندگان

نویسنده: محمد شیران
تاریخ: ۰۸:۱۸:۰۷ ۱۳۹۱/۰۱/۲۷

آقای نصیری واقعا از شما سپاسگذاریم. میدونیم که جدا از تجربیات ارزندتون که ساعتها برای کسبش وقت و تلاش گذاشتین نوشتن هر کدوم 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: ۲۳:۵۱:۳۱ ۱۳۹۱/۰۱/۲۹

ممنون از مطالب عالی
آیا این امکان هست داخل پوشه های مهم (پوشه هایی که توسط Areas تعریف شد)که نیاز به لاگین 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: ۰۹:۲۶:۳۹ ۱۳۹۱/۰۲/۰۱

در فایل کانفیگ اصلی برنامه (در ریشه اصلی سایت) می‌تونید با مقدار دهی تگ location اینکار رو انجام بدید. ولی روش توصیه شده‌ای برای ASP.NET MVC نیست. استفاده از فیلتر Authorize روش امن‌تری است چون مستقل از تعاریف مسیریابی عمل می‌کند و امکان فراموشی آن پس از تغییرات در مسیریابی نیست.

نویسنده: asrin
تاریخ: ۱۸:۴۰ ۱۳۹۱/۰۶/۲۹

من دارم رو یه پروژه بزرگ با MVC کار می‌کنم.می خواستم ازتون بپرسم که گذاشتن این فیلترها در برنامه ضروریه؟

نویسنده: وحید نصیری
تاریخ: ۱۹:۱۲ ۱۳۹۱/۰۶/۲۹

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 برنامه‌ای ضروری است.

نویسنده: sara zarmehr
تاریخ: ۲۱:۳۷ ۱۳۹۱/۱۰/۱۷

فیلتر سفارشی رو چگونه ساختید که namespace این گونه نوشته شده؟

```
namespace MvcApplication12.CustomFilters
{
    .....
    .....
}
```

نویسنده: وحید نصیری
تاریخ: ۲۱:۴۴ ۱۳۹۱/۱۰/۱۷

داخل یک پوشه به نام CustomFilters. سورس‌های کامل این سری در دسترس است .

عنوان: ASP.NET MVC #16
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۲۷ ۲۳:۴۸:۰۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

مدیریت خطاها در یک برنامه ASP.NET MVC

استفاده از فیلتر HandleError

یکی از فیلترهای توکار ASP.NET MVC به نام HandleError، می‌تواند کار هدایت کاربر را به یک صفحه‌ی خطای عمومی، در حین بروز استثنایی 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 نمائید:

```
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 آن‌را خارج از دیباگر VS.NET اجرا کرد (آدرس برنامه را مستقیما خارج از VS.NET در یک مرورگر وارد کنید). همچنین یک سطر زیر را نیز لازم است به فایل web.config برنامه اضافه نمائید:

```
<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 استثنا در متد 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 متد Application_Start فایل Global.asax.cs اضافه نمائید:

```
GlobalFilters.Filters.Add(new HandleErrorAttribute());
```

البته نیازی به انجام اینکار نیست زیرا اگر به متد RegisterGlobalFilters 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 Views\Shared\CustomErrorView.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 متد نیز وجود دارد:

```
[HandleError(ExceptionType = typeof(NullReferenceException), View = "ErrorHandling")]
```

دسترسی به اطلاعات استثناء در صفحه نمایش خطاها

زمانیکه برنامه به صفحه خطا 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 صرفا به عنوان اطلاعات عمومی در نظر داشته باشید. زیرا اگر قرار باشد مجددا اصل استثناء را نمایش دهیم، همان صفحه زرد رنگ ASP.NET شاید بهتر باشد.

استفاده از تگ customErrors در فایل Web.config برنامه

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 خطا را 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 حین توزیع برنامه باید بکارگرفته شود.

مدیریت خطاهای رخ داده خارج از MVC Pipeline

HandleErrorAttribute تنها استثناهای رخ داده داخل ASP.NET MVC Pipeline را مدیریت می‌کند (یا خطاهایی از نوع 500). اگر این نوع استثناها خارج از آن رخ دهند مثلا 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 متدی به نام notfound هدایت خواهد شد. بنابراین نیاز به کنترلر زیر وجود دارد؛ به علاوه به ازای هر متد هم یک View متناظر باید اضافه شود (کلیک راست روی نام متد 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 مثلا آدرس غیرموجود http://localhost/xyz را وارد کنید.

استفاده از فیلتر HandleError اجباری نیست

در همین قسمت قبل پس از افزودن customErrors و defaultRedirect آن 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 متد Index کنترلر Error اجرا خواهد شد. به عبارتی الزاما نیازی به استفاده از فیلتر HandleError نیست و به کمک مقدار دهی صحیح تگ customErrors، کار نمایش خودکار صفحه سفارشی خطاها به کاربر انجام خواهد شد.

البته بدیهی است که گزینه‌های نمایش یک View خاص به ازای استثنایی 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 جزئیات 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 است و اصلا 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 نمائید و یا به کمک NuGet هم به سادگی قابل دریافت است. پس از دریافت، ارجاعی را به اسمبلی آن (Elmah.dll) اضافه نمائید. 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 نمائید. مثلا 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 خطایی 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 نمائید. به این ترتیب امکان 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 لاگ شده به همراه جزئیات کامل آن‌ها مراجعه کنید.

مشکل: استثناهای برنامه توسط ELMAH لاگ 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 متد RegisterGlobalFilters فایل Global.asax.cs 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

HandleError توکار) را به موتور ELMAH هدایت کند. سایر خطاهای مدیریت نشده به صورت خودکار توسط ELMAH ثبت خواهند 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!

حین استفاده از ELMAH یک نکته را فراموش نکنید:
اگر allowRemoteAccess آن‌را به عدد 1 تنظیم کرده‌اید، به هیچ عنوان از نام پیش فرض elmah.axd استفاده نکنید (هر نام اختیاری 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- حتما قسمت customErrors فایل وب کانفیگ برنامه را دهی کنید (این مورد را به چک لیست اجباری تهیه یک برنامه ASP.NET MVC اضافه کنید).
3- استفاده از فیلتر HandleError اختیاری 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 رخ داده در برنامه، از ELMAH استفاده کنید و بی‌جهت وقت خودتان را صرف بازنویسی این افزونه ارزشمند نکنید.

مطالب مشابه

معرفی ELMAH
ثبت استثناهای مدیریت شده توسط ELMAH

نظرات خوانندگان

نویسنده: Ahmadi rad
تاریخ: ۲۱:۴۵:۴۰ ۱۳۹۱/۰۱/۲۸

سلام علیکم
از غیرتی که در یاد آموزی مطالب در سطح خود به زبان فارسی دارید از شما سپاسگذاریم.
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 پسندیده ای است.

نویسنده: Salehi
تاریخ: ۱۰:۲۱:۳۱ ۱۳۹۱/۰۱/۳۰

با سلام و تشکر واقعی از این مطالب مفید.
اگه از روش اول برای ثبت استثناها توسط Elmah استفاده شود، یعنی حذف یک سطر از فایل Global.asax.cs، فقط استثناها ثبت می شود ، یا صفحه خطایی هم نمایش داده می شود؟

نویسنده: وحید نصیری
تاریخ: ۰۹:۲۳:۲۹ ۱۳۹۱/۰۲/۰۱

بستگی داره cutomErrors رو مقدار دهی کردید یا خیر. اگر بله، مطابق مقدار دهی آن رفتار خواهد شد؛ مثلا یک صفحه سفارشی نمایش داده می‌شود. ولی خطایی از دست ELMAH در نمیره!

نویسنده: فربد صالحی
تاریخ: ۱:۹ ۱۳۹۱/۰۴/۰۲

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 انجام شد اما onexception در اونجا هم فراخوانی نمی‌شد.

3- آیا راهی وجود داره که برای هر area بشه مسیر اختصاصی برای خطاهای 403و 404 معرفی کرد؟ چون در بخش customerrors یک مسیر کلی داده میشه

4- این ادیتوری که شما اینجا استفاده کردید خیلی سبک و قویه. اگه میشه معرفیش کنید!

نویسنده: شاهین کیاست
تاریخ: ۴:۳۹ ۱۳۹۱/۰۴/۰۲

ادیتور http://redactorjs.com/ هست.

نویسنده: وحید نصیری
تاریخ: ۸:۴۹ ۱۳۹۱/۰۴/۰۲

- 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 رو تبدیل به حالت Release می‌کنه اما به این معنا نیست که مکانیزم‌های ASP.NET رو کلا از کار می‌اندازه. مصرف حافظه برنامه کمتر میشه. debug symbols از اسمبلی نهایی حذف خواهند شد. کامپایلر روی کد نهایی بهینه سازی‌هایی در جهت بالابردن سرعت انجام می‌ده.
- در MVC اگر از فیلتر Handle Error استفاده بشه application_error فراخوانی نخواهد شد. به علاوه با وجود ELMAH ضرورتی به استفاده از Handle error و application_error نیست. توضیح دادم در متن فوق.
ضمن اینکه دیباگ کار شخصی افراد نیاز به بودن پروژه آن‌ها و بررسی آن(ها در یک انجمن 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.
- custom errors قرار گرفته در ریشه سایت بر روی کل سایت اعمال می‌شود (و تمام برنامه‌هایی که در زیر پوشه‌های آن هستند). در فایل‌های کانفیگ ASP.NET مباحث ارث بری وجود دارند. برای لغو آن یکی از راه‌ها استفاده از تگ location است، مثلا:

```
<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 خطاها 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 بهتر است از Elmah.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>
```

ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۲۳ ۱۸:۵

- فرقی نمی‌کنه. Elmah.MVC توضیحات این قسمت رو خلاصه کرده.

- اگر به استفاده کنندگان اطمینان دارید، مهم نیست.

نویسنده: امیر
تاریخ: ۱۳۹۱/۰۶/۰۲ ۱۵:۲۰

سلام
اگر تو سایتمون از area استفاده کرده 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
ممنون.

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 HandleErrorAttribute 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: ۱۳۹۲/۱۱/۲۲ ۹:۲۳

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 بی‌جهت وقت خودتان را صرف بازنویسی این افزونه ارزشمند نکنید ...»

نویسنده: داریوش حمیدی
تاریخ: ۱۳۹۳/۰۷/۱۵ ۹:۴۰

سلام؛ اگر امکان داره Sample این آموزش را بزارید ... من در ثبت خطاها در فایل xml دچار مشکل شده ام ..
ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۱۵ ۹:۴۵

دریافت تمام مثال‌های این سری: MVC_Samples

نویسنده: حمید کشاورز
تاریخ: ۱۳۹۳/۰۷/۱۶ ۱۱:۴۷

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 بدهم...
ممنون //

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۱۶ ۱۱:۵۸

فقط تنظیمات فایل web.config را اعمال کنید. مابقی مسایل آن خودکار است.

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۳/۰۸/۱۵ ۱۸:۴

با سلام و تشکر از مطلب خوب شما.
در محیط لوکال خطاها توسط Elmah به خوبی لاگ می‌شود ولی بعد از پابلیش به سرور هیچ خطایی لاگ نمی‌شود. تمام تنظیمات بالا را نیز خط به خط رعایت کردم. در سمت سرور به پوشه ErrorLogs نیز پرمیژن لازم برای read و write دادم ولی جوابی نگرفتم. علت چه می‌تواند باشد. هاست هم یک هاست اشتراکی است. با تشکر.

عنوان: ASP.NET MVC #17
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۲۹ ۰۸:۲۵:۰۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

فیلترهای امنیتی ASP.NET MVC

ASP.NET MVC 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

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 صرفا 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 ایی که معرفی شد، بر روی تمام خواص شیء BlogPost اعمال می‌شود. اما اگر فقط بخواهیم که مثلا 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 ایی 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 صریحا آن‌را تبدیل به HTML کنیم (مثلا استفاده از متد Html.Raw). به عبارتی خروجی Razor در حالت پیش فرض در مقابل حملات XSS مقاوم است مگر اینکه آگاهانه بخواهیم آن‌را غیرفعال کنیم.

مطلب تکمیلی
مقابله با XSS ؛ یکبار برای همیشه!

فیلتر RequireHttps

به کمک ویژگی یا فیلتر RequireHttps، تمام درخواست‌های رسیده به یک متد 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 CSRF یا Cross site request forgery 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 لاگین 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 مثلا 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 نبسته است، «احتمال» اجرا شدن درخواست مهاجم بالا است (خصوصا اگر از مرورگرهای قدیمی استفاده کند).
بنابراین نیاز است بررسی شود آیا درخواست رسیده واقعا 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 خطاها ابتدا باید داخل فرم‌های برنامه از متد Html.AntiForgeryToken استفاده کرد. کار این متد ایجاد یک فیلد مخفی با مقداری منحصربفرد 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 متد 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 استثنا صادر شده و از پردازش اطلاعات رسیده جلوگیری می‌شود.
علاوه بر این‌ها بهتر است حین استفاده از متد 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، آن‌را 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 AntiForgeryToken را به چک لیست اجباری تولید تمام فرم‌های برنامه اضافه نمائید.

مطلب مشابه
Anti CSRF module for ASP.NET

فیلتر سفارشی بررسی Referrer

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 UrlReferrer 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 خواهند شد. مثلا در متد Upload قسمت 11 این سری، منعی 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 aspx یا cshtml یا هر فایل قرار گرفته در پوشه Views را مستقیما در مرورگر خود وارد کند، با پیغام HttpNotFound مواجه 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 فریم ورک‌ها برفراز آن معنا پیدا می‌کنند.

بنابراین در پوشه فایل‌های آپلودی به سرور خود یک web.config را با محتوای فوق ایجاد کنید (و فقط باید مواظب 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 مثلا 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 نوشتن این آموزش‌ها تا قسمت هفدهم.
من سر بخش آپلود فایل‌ها یه سوال دارم.
در زمینه هکینگ و ویروس‌نویسی تقریبا 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 ویا خراب‌شدن سرور میشن. درباره‌ی درست‌بودن این زیاد مطمئن نیستم ولی خوب، این ممکنه خودش یه مشکل امنیتی باشه.

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 ؟

ممنون

نویسنده: Salehi
تاریخ: ۱۰:۵۲:۳۹ ۱۳۹۱/۰۱/۳۰

-1 جایی خونده بودم افزونه هایی برای بعضی از مرورگرها وجود داره که می تونه UrlReferrer رو به دلخواه کاربر تغییر بده و بنابراین استفاده از این ویژگی کمکی نمی کنه.درسته؟
-2 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 برنامه های تحت وب و نه در وبسایت ها

نویسنده: وحید نصیری
تاریخ: ۰۹:۱۹:۰۹ ۱۳۹۱/۰۲/۰۱

- درسته. ولی محض احتیاط بررسی این مطلب ضرری نداره.
- بله. در برنامه‌های وب اصلا چنین اجازه‌ای رو نباید داد.

نویسنده: وحید نصیری
تاریخ: ۰۹:۲۰:۱۷ ۱۳۹۱/۰۲/۰۱

این یک کار سورس باز هست. می‌تونید 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 (مثلا پوشه آپلود) حذف می‌کنه.

نویسنده: asrin
تاریخ: ۱۲:۵۵ ۱۳۹۱/۰۶/۳۰

اگر داخل فرم اطلاعاتمان را با ajax به اکشن‌ها فرستاده باشیم چطور 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
نمیسازه که همراه ان guid که browser 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 یوز کنم در حالی که DataLayer من در یک پروژه و Dll مجزا است .

یوز کردن یا Refrence 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 EntityFrameWorkو دادها یوز 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 ViewModel 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 میگره 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: ۱۳۹۲/۰۹/۱۱ ۹:۰

ذکر دستی Salt از MVC4 به بعد حذف شده. این مورد به صورت خودکار محاسبه و اعمال می‌شود.

نویسنده: لیلا
تاریخ: ۱۳۹۲/۰۹/۱۵ ۲۲:۵۲

با تشکر
به چه روشی میتوان از AntiForgeryToken در Ajax.ActionLink استفاده کرد ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۱۶ ۱۳:۹

برای ارسال صحیح anti forgery token در حین اعمال Ajax 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: ۱۳۹۲/۱۱/۲۱ ۱۵:۵۶

سلام

من مرتبا با خطای AntiForgeryToken مواجه میشم.

یکبار این پیغام صرفا بر روی host ایجاد میشه وگاهی 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 خطا چیست و چرا همیشگی نیست و مشخص کردن کد و الگوریتم کار صحیحی هست یا خیر؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۲۱ ۱۶:۴۷

چه خطایی دقیقا گرفتید؟ (من تابحال مشکلی نداشتم با فیلتر آنتی‌فورجری)

نویسنده: رضا گرمارودی

تاریخ: ۱۷:۱۷ ۱۳۹۲/۱۱/۲۱

عنوان خطا: 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 Vs2013)

موارد مشابه در سایر انجمن‌ها : 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 خطا نمایش داده نمی‌شود. چرا پیغام نمی‌توانم Decrypt کنم؟ پس چرا در همین برنامه ولی دربخش دیگری که از anti-forgery token استفاده شده مشکل وجود ندارد؟

نویسنده: وحید نصیری
تاریخ: ۱۸:۷ ۱۳۹۲/۱۱/۲۱

خلاصه موارد ممکن:

- در یک صفحه چندبار از Html.AntiForgeryToken استفاده شده‌است. این مورد کوکی آنتی‌فورجری را تخریب می‌کند و نهایتا اطلاعات آن قابل رمزگشایی و مقایسه در سمت سرور نخواهد بود.

- تنظیم AntiForgeryConfig.SuppressIdentityHeuristicChecks = true را به فایل global.asax.cs نیز اضافه کنید.

آنتی‌فورجری توکن وضعیت کاربر لاگین شده به سیستم را نیز نگهداری و رمزنگاری می‌کند. در این حالت اگر در یک برگه‌ی دیگر لاگ آف کنید و در برگه‌ی قبلی سعی در ارسال فرم، آنتی‌فورجری توکن یک پیام خطا را نمایش می‌دهد. با تنظیم SuppressIdentityHeuristicChecks = true این بررسی وضعیت لاگین 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: ۱۹:۴۶ ۱۳۹۲/۱۱/۲۱

با سلام و عرض خسته نباشید.

من در پروژه ام به شکل ذیل عمل کردم؛ آیا اشتباه 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؛ چه درخواست نمایش یک View باشد چه 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: ۱۳۹۲/۱۱/۲۱ ۲۳:۹

سلام؛ ممنون بابت پاسختون. این طوری به forgeryToken نگاه نکرده بودم.

در صفحه برای Delete و ثبت اطلاعات و یا حتی فیلترکردن اطلاعات و هرجا که کاربر اطلاعاتی وارد می‌کند از AntiForgeryToken استفاده کردم و حذف و ورود اطلاعات و فیلتر کلا در یک صفحه اما هرکدام در تگ Form مجزایی ارسال و دریافت می‌شود و در هر تگ Form یک ForgeryToken گذاشتم.

برای اونها چه کار کنم؟ اصلا کار بنده درست بوده؟ من هرکجا کاربر اطلاعاتی وارد کرده از Token استفاده کردم ولی مشکل اینجاست که با توجه به طراحی کل اونها نیازه که در یک صفحه باشه.

می تونم یک AntiForgeryToken کلی داشته باشم و با جاوا اسکریپت اون و بخونم و به Form.Serilaise() اضافه کنم. اما می‌خواستم راه بهینه اون و بپرسم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۲۱ ۲۳:۴۱

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 }), // اضافه کردن توکن
   dataType: "html", // نوع داده مهم است
   // .....
});
```

سه نکته اینجا مهم است:

- data type حتما باید در این حالت html باشد.
- در قسمت data متد addToken کار افزودن خودکار آنتی فورجری‌توکن 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 ۱۳:۱۹

با سلام
من از روش ایجاد فایل web.config برای upload فایل استفاده کردم. جواب هم داد. ولی مشکل این هست که در داخل خود صفحات هم به آن فایل‌ها دسترسی پیدا نمیکنه.
اون صفحه خطای عدم اجازه جهت دسترسی به پوشه ای که مستقیم در مرروگر وارد شده را چه طور سفارشی کنم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۰۶ ۱۳:۴۵

- مدیریت خطاها در یک برنامه ASP.NET MVC
- محدود کردن کاربرها به آپلود فایل‌هایی خاص در ASP.NET MVC
- کنترل عمومی فایل‌های آپلودی در ASP.NET MVC

نویسنده: امیر نوروزیان
تاریخ: ۱۳۹۳/۱۰/۱۱ ۹:۲

با سلام
۱- با توجه به اینکه برای این فیلتر ValidateAntiForgeryToken باید حتما کوکی فعال باشه آیا درسته استفاده کنیم. اگه خواسته نباشیم از این فیلتر استفاده کنیم روشی دیگری هست.
۲- بین Referrer و ValidateAntiForgeryToken کدوم فیلتر استفاده شود. توصیه به استفاده کدوم فیلتر. آیا همزمان میشه استفاده کرد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۱۱ ۱۰:۱۰

- 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 صرفا 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 عمده‌ای 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).
- AntiForgeryToken مهم‌تر است. امکان استفاده‌ی همزمان هم وجود دارد. نمونه‌ی استفاده از چند ویژگی در یک اکشن متد، ذیل فیلتر مربوط به آن، در متن هست.

عنوان: ASP.NET MVC #18
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۳۱ ۱۰:۴۵:۰۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

اعتبار سنجی کاربران در ASP.NET MVC

دو مکانیزم اعتبارسنجی کاربران به صورت توکار در ASP.NET MVC در دسترس هستند: Forms authentication و Windows authentication.

در حالت Forms authentication، برنامه موظف به نمایش فرم لاگین به کاربرها و سپس بررسی اطلاعات وارده 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 اینترانت‌های 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 منعی هم برای استفاده در حالت اینترانت ندارد).

ضمنا 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 Authorize در ASP.NET MVC

یکی دیگر از فیلترهای امنیتی ASP.NET MVC به نام Authorize، کار محدود ساختن دسترسی به متدهای 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 آن‌را نخواهند داشت. فیلتر Authorize همواره قبل از تمامی فیلترهای تعریف شده دیگر اجرا می‌شود.

فیلتر Authorize با پیاده سازی اینترفیس System.Web.Mvc.IAuthorizationFilter توسط کلاس System.Web.Mvc.AuthorizeAttribute در دسترس می‌باشد. این کلاس علاوه بر پیاده سازی اینترفیس یاد شده، دارای دو خاصیت 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 تنهایی بکارگرفته 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 متد 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 یا همان Forms authentication تنظیم شده‌اند. در اینجا کلیه کاربران اعتبار سنجی نشده، به کنترلری به نام Account و متد 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 forms آن‌را 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 ایی 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 لاگین نیز 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 پوشه Content از سیستم اعتبارسنجی اجباری خارج می‌شود و تمام کاربران به آن دسترسی خواهند داشت.
به علاوه امکان 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 استفاده شود). 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، حتما 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 صریحا 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 ASP.NET MVC 4 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 نمائید.

البته نوشتن فیلتر سفارشی AllowAnonymousAttribute در ASP.NET MVC 3.0 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 اکنون 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 آن‌را با ویژگی Authorize، مزین نمائید. 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 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 مراجعه کرده و فایل applicationhost.config را باز نمائید. تگ windowsAuthentication را یافته و ویژگی enabled آن‌را که false است به true تنظیم نمائید. اکنون اگر برنامه را مجددا 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 الزامی 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 نمائید.
سپس نیاز است نکته «تنظیمات اعتبار سنجی اجباری تمام صفحات سایت» را به فایل وب کانفیگ برنامه اعمال نمائید تا نیازی نباشد فیلتر Authorize را در همه جا معرفی کرد. سپس نحوه معرفی پیش فرض Forms authentication تعریف 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 لاگین برنامه اشاره می‌کند. defaultUrl مسیری است که کاربر پس از لاگین به صورت خودکار به آن هدایت خواهد شد. همچنین نکته‌ی مهم دیگری را که باید رعایت کرد، name ایی 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 منحصربفرد انتخاب کنید، در غیراینصورت تداخل رخ داده و گزینه مرا به خاطر بسپار شما کار نخواهد کرد.

کار slidingExpiration که در اینجا تنظیم شده است نیز به صورت زیر می‌باشد:

اگر لاگین موفقیت آمیزی ساعت 5 عصر صورت گیرد و timeout شما به عدد 10 تنظیم شده باشد، این لاگین به صورت خودکار در 5:10 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 5:15 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 پوشه 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 وهله‌ای از شیء Account، یک رشته را به نام returnUrl نیز تعریف 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 تعریف 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 تعاریف 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 faild! 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 متد 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 متد LogOn از نوع HttpPost، کار اعتبارسنجی 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 متد RedirectFromLoginPage کاربر را به صفحه پیش فرض سیستم هدایت می‌کنیم. یا اگر returnUrl ایی وجود داشت، آن‌را پردازش خواهیم کرد.

برای پیاده سازی خروج از سیستم هم تنها کافی است متد FormsAuthentication.SignOut فراخوانی شود تا تمام اطلاعات سشن و کوکی‌های مرتبط، به صورت خودکار حذف گردند.

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 IsUserInRole 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

نویسنده: Info
تاریخ: ۱۳۹۱/۰۱/۳۱ ۱۵:۵۴:۲۱

با سلام و درود و تحیتواقعا دست مریزاد جناب مهندسدر فیلتر [("Authorize(Roles="Admin")] آیا میشود را بصورت دینامیک تعیین کرد به نحوی که مثلا از بانک بشود خواندباتشکر

نویسنده: Msafdel
تاریخ: ۱۳۹۱/۰۱/۳۱ ۱۷:۱۵:۳۰

ممنون آقای نصیری. این سری آموزشها خیلی عالی بود و استفاده کردم.

نویسنده: سعید شیرزادیان
تاریخ: ۱۳۹۱/۰۱/۳۱ ۲۲:۰۲:۰۳

استاد نصیری عزیز سلام
مطالب در رابطه با Mvc بسیار جالب و مفید می باشد من گام به گام از مرحله اول تا 11 را تمرین کردم. اگر امکان دارد فراخوانی اطلاعات از دیتابیس و استفاده از آنها نیز اشاره ای بکنید.

با تشکر - شیرزادیان

نویسنده: 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 تره؟

نویسنده: مجتبی حسینی
تاریخ: ۱۳۹۱/۰۲/۰۱ ۰۱:۴۷:۰۹

با سلام و درود و تحیت
واقعا دست مریزاد جناب مهندس
در فیلتر [("Authorize(Roles="Admin")] آیا میشودRole را بصورت دینامیک و پویا تعیین کرد به نحوی که مثلا از بانک خوانده شود
باتشکر

نویسنده: 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 خودکار با الگوریتم‌های پردازش موازی براش نیست.

نویسنده: 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 آن‌را پویا تعریف کرد. اما می‌شود با ارث بری AuthorizeAttribute : CmsAuthorizeAttribute و ایجاد یک فیلتر سفارشی اینکار رو انجام داد. بعد در متد public override virtual bool AuthorizeCore فرصت خواهید داشت با بانک اطلاعاتی کار کنید.

نویسنده: Salehi
تاریخ: ۱۳۹۱/۰۲/۰۱ ۱۲:۵۹:۵۴

چطور میشه اسم controller، area و action رو به این تابع فرستاد؟ جستجو کردم ولی چیزی پیدا نکردم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۰۱ ۱۳:۲۶:۵۴

- از کلاس AuthorizeAttribute ارث بری کنید. بعد داخل آن یک خاصیت به نام مثلا public string AreaName تعریف کنید. این ویژگی سفارشی اکنون می‌تواند از پارامتر AreaName هم استفاده کند و استفاده داخلی از آن با تحریف متد AuthorizeCore میسر خواهد شد.
- اگر متد OnAuthorization را تحریف کنید، به filterContext.Controller دسترسی خواهید داشت.
- ضمن اینکه شما در سازنده این کلاس فیلتر سفارشی، فرصت مقدار دهی خواصی مانند Roles را بر اساس اطلاعات بانک اطلاعاتی خواهید داشت. یعنی به این شکل هم می‌شود آن‌را پویا تعریف کرد.
- توسط HttpContextBase httpContext متدهای تحریف شده به اطلاعات کاربر جاری می‌شود دسترسی یافت (httpContext.User.Identity.Name).

نویسنده: Salehi
تاریخ: ۱۳۹۱/۰۲/۰۱ ۲۰:۱۸:۴۱

این واقعا خیلی عالی بود. متشکرم.
پویا کردن AuthorizeAttribute واسه من جای سوال داشت که شما برطرفش کردید.
به نظرم استفاده از Attribute ها واقعا تو mvc خیلی کاربرد داره و خیلی کارها رو راحت میکنه. ولی متاسفانه 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 .

نویسنده: 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 متد

```
ShouldRedirect(returnUrl)
```

هم مقدار true رو بر می‌گردونه و نتیجه اینکه دوباره بر می‌گرده به /User/Create و اونجا هم دوباره برمی گرده به همین اکشن و این Loop تکرار می‌شه.

این کد برای من اینجوری کار می‌کنه

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۱۷ ۲۲:۴۴

علتش می‌تونه عدم دریافت نقش‌های کاربر لاگین کرده به سیستم باشد. اگر در قسمت ایجاد کاربر، نقش مورد نیاز، مدیریتی است و مدیر (شخص لاگین کرده) به این صفحه هدایت شده و باز به لاگین بر می‌گردد، یعنی شخص لاگین شده دارای این نقش نیست. یا اگر این دسترسی تعریف شده، قسمت پروایدرنقش‌های سفارشی، درست عمل نکرده.
ضمن اینکه به نظر این redirectها 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 رو اینجوری تغییر دادم الان مشکل وقتی پیش میاد که کاربری که Admin نیست قبلا Login کرده و می‌خواد به اون صفحه دسترسی پیدا کنه.در بقیه موارد درست کار می‌کنه

```
<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

ممنون. 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: ۱۳۹۱/۰۴/۱۸ ۲۳:۴۷

ممنون

نویسنده: دانشجو
تاریخ: ۱۳۹۱/۰۶/۰۱ ۱۲:۴۶

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: ۱۳۹۱/۰۶/۰۱ ۱۳:۲۶

توضیح داده شد. throw new Exception (در SiteAuthorizeAttribute یاد شده در نظر فوق) سبب می‌شود تا این redirectها متوقف شود و همچنین دسترسی غیرمجاز نیز لاگ شود در سیستم.

نویسنده: دانشجو
تاریخ: ۱۳۹۱/۰۶/۰۱ ۱۵:۱۲

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 اصلا اجرا نمی‌شود؛
آیا SiteAuthorizeAttribute باید در GlobalFilterCollection اضافه شود؟ یا...

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۰۱ ۱۶:۰

احتمالا **Role provider سفارش ی** شما درست ثبت نشده و کار نمی‌کند. در این مورد در انتهای متن قسمت جاری بحث شده.

نویسنده: Naser Tahery
تاریخ: ۱۳۹۱/۰۶/۲۶ ۰:۴۲

سلام آقای نصیری
با توجه به این نوع پیاده سازی[لایه دسترسی به داده‌ها توسط service layer] (+) اگر خواسته باشیم نقش‌های یک کاربر را

بدست بیاوریم، باید از لایه‌ی 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 میومد. 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; //نام نقش
                if (roles != null)
                {
                    return roles.ToArray();
                }
            }

            return new string[] {};
        }
}
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۲۶ ۱:۲

CustomRoleProvider چون مستقیما و راسا توسط خود ASP.NET وهله سازی می‌شود، وارد پروسه متداول Controller Factory 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<IRolesService>();
            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.Rol;
                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 وهله سازی PublishingContext نیست چون الگوی واحد کار را نقض می‌کند؛ چون هر وهله 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. مابقی مسایل تفاوتی نمی‌کند.

نویسنده: رضا بزرگی
تاریخ: ۱۳۹۱/۰۷/۰۶ ۲۱:۴۵

برای یکپارچه‌سازی membership توکار ASP.NET با برنامه‌ی خودمون چه‌کار 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

ممنونم بابت معرفی این مورد.

این مهم در این سری آموزشی‌های سایت نیست. چون زمان نگارش اصلا SimpleMembership وجود نداشته.
به‌طور خلاصه توضیحاتی بدم:
استفاده از این پروایدر به صورت توکار فقط در تمپلیت Internet Application استفاده شده. خاصیت این سیستم این‌است که سختی آن سیستم قدیمی که از ASP.NET 2.0 باب شد را ندارد و واقعا 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>
```

SimpleMembership به شما اجازه میدهد با برنامه‌ی خودتان در یک پایگاه داده ذخیره کنید. و در واقع یکپارچه کنید. برای این کار در 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 فیلدهای مربوطه را اضافه کنید و همچنین data anotationهای 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 متد Register با خاصیت HttpPost باید WebSecurity.CreateUserAndAccount را به‌درستی مقدار دهی کرد. چندین overload دارد. در overloadی که دارای قسمت object است میتوانیم از یک Anonymus 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: ۱۳۹۱/۰۷/۱۷ ۱۷:۵۴

سلام؛
User.Identity.IsAuthenticated همیشه مقدار false رو بر میگردونه دلیلش چیه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۱۷ ۱۸:۱۴

- 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 (متد LogOn) نیاز خواهد بود تا کوکی لازم و RedirectFromLoginPage صحیحی اعمال شود.

نویسنده: mina
تاریخ: ۱۳۹۱/۰۷/۲۶ ۹:۵۷

```
<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
ممنون از مطالب خوبتون

من تو این قسمت مشکل دارم تا اینجا درست کار میکنه که کسانی که authenticate نکردن برن به صفحه لاگین اما cssها رو با اینکه location هم گذاشتم نمیاره
بالای system.web هم می‌زنم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۲۶ ۱۰:۵۹

- در این تنظیمات فرض بر این است که پوشه content محل قرارگیری فایل‌های css است و همچنین این پوشه 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 نظرگرفتم 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 کنترلرها . 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 .

ممنون 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 مستقیما استفاده نکنید. روش پیچ و تاب آن باشد بر اساس روشی که انتخاب می‌کنید.
- من از روش ترکیبی استفاده می‌کنم. حتما هم استفاده می‌کنم. چون دقیقا 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 برمبنای هش‌های عمومی هستند. گیرم حتی اگر کار شما سورس باز باشد، تبدیل الگوریتم‌های ترکیبی آن به الگوریتم‌های بهینه قابل اجرای بر روی GPU کار هرکسی نیست.

استفاده مستقیم از الگوریتم‌های هش‌های متداول در یک پروژه عمومی با تعداد کاربر بالا = عین حماقت

نویسنده: احمد احمدی
تاریخ: ۱۳۹۲/۰۴/۰۷ ۰:۱۲

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.
اصلا چرا نمیشه درخواست به این محتوا رو به یک کنترلر هدایت کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۰۷ ۰:۳۸

- برای اینکه عموما 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 لاگین 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 pipline باشد، به صورت پیش فرض در وب کانفیگ runAllManagedModulesForAllRequests وجود دارد (یعنی در این حالت تمام فایل‌ها از موتور ASP.NET رد خواهند شد). در IIS6 است که یا باید دستی تنظیمات را تغییر داد یا کدنویسی کرد. ضمن اینکه مدیریت این مسایل توسط IIS سربار 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 ۱۵:۱۰

با سلام.
چگونه میتوان برای قسمت admin سایت از یک فرم لاگین و برای قسمت‌های دیگر از فرم لاگین دیگر استفاده کرد. تنظیمات آن در وب کانفیگ چگونه است؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۲۴ ۱۶:۴۳

بر اساس نقش‌های کاربران پس از لاگین، 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 ۱۷:۴۴

سپاس از پاسخ شما.
ولی منظورم ، داشتن چند فرم لاگین 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 لاگین با ورودی‌های نام کاربری ، پسوورد دارند. با تشکر فراوان.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۲۴ ۱۸:۲۰

در سؤال قبل تنظیمات وب کانفیگ رو می‌خواستید برای چند لاگین. در وب کانفیگ، کار خاصی انجام نمیشه. اونجا فقط مشخص میشه که طول عمر کوکی لاگین چند روز باشه یا پس از لاگین، 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 Roleهای مشخص سیستم. اگر زیر سیستمی باید صرفا 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 لاگین دوم، تنظیم شده یا خیر (یا در ساده‌ترین حالت از سشن استفاده کنید). اگر خیر، بر اساس Role مشخص صفحه جاری، به یک صفحه لاگین ثانویه هدایت شود تا کاربر بتواند کوکی یا سشن لازم را پس از لاگین 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، مثلا 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 خواهند بود مفهوم دارد.
اما اگر سایتی پویا طراحی شده و از روز اول طرح دقیقی ندارد، مثلا در آن صفحات مختلفی به صورت پویا اضافه می‌شوند و قرار است به هر صفحه‌ای نقش(های) خاصی انتساب داده شوند که از روز اول پیش بینی و طراحی نشده، فقط کافی است همین فیلتر AuthorizeAttribute را با ارث 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 لاگین شده به سیستم (filterContext.HttpContext.User) می‌تواند به این صفحه دسترسی داشته باشد یا خیر و هدایتش کند (filterContext.HttpContext.Response.Redirect) به صفحه لاگین برنامه (FormsAuthentication.LoginUrl). یا حتی در اینجا بر اساس نام کنترلر و اکشن متد 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 کنترل و Home استفاده کردم اما متعجبم که فقط روی کنترل Account کار می‌کند و در کنترل Home بازهم 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 نمائید تا نیازی نباشد فیلتر Authorize را در همه جا معرفی کرد. » کد زیر را به فایل کانفیگ (بعلاوه 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 کریمی
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 متدی 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: محمدرضا 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 می‌شود ...

```
[HttpPost]
        public virtual ActionResult LogOn(Accounts acc)
        {
            FormsAuthentication.SetAuthCookie(acc.Username, acc.RememberMe);
            return View();
        }
```

لطفا 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: ۱۳۹۲/۱۱/۰۳ ۹:۹

- نیازی نیست تمام متدهای RoleProvider دات نت پیاده سازی شوند. برای یک برنامه پیاده سازی دو متد IsUserInRole، 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 برنامه مطابق معمول توسط DbSetها در معرض دید EF قرار می‌دهید. مابقی آن کارکردن معمولی با این دو جدول اضافه شده به برنامه است:

```
public class EfRolesService : IRolesService
    {
        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 لاگین، user.Id در FormsAuthentication.SetAuthCookie تنظیم می‌شود ؛ یعنی userName در این RoleProvider به id 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<IRolesService>();
            return rolesService.IsUserInRole(username.ToInt(), roleName);
        }

        public override string[] GetRolesForUser(string username)
        {
            var rolesService = ObjectFactory.GetInstance<IRolesService>();
            return rolesService.GetRolesForUser(username.ToInt());
        }
// مابقی نیازی نیست پیاده سازی شوند
```

نویسنده: رضا منصوری
تاریخ: ۱۳۹۲/۱۱/۰۳ ۹:۲۷

با سلام؛

User.Identity.Name وقتی که RedirectToAction به همون کنترلی که هستیم انجام میدیم کار نمیکنه؟

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 رو نال بر میگردونه! بقیه جاها از User.Identity.Name استفاده میکنم و هیچ مشکلی نیست!

با تشکر

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۲/۱۱/۰۳ ۹:۴۵

ممنون، عالیه

فقط این ToInt به چه صورت username را به int تبدیل میکنه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۰۳ ۱۰:۵

یک متد الحاقی است به این شکل:

```
public static int ToInt(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(...
```

ضمنا این رشته در MVC اگر کاربر لاگین نکرده باشد، صرفا string.Empty خواهد بود و نه null (با یک برنامه ساده و جدید آن‌را امتحان کنید). بنابراین مشکل از قسمت دیگری از کدهای شما ناشی می‌شود. stack trace را نیاز است دقیقا بررسی کنید.

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۲/۱۱/۰۳ ۱۶:۴۰

خیلی ممنون،

User.Identity.Name مقدار صفر رو برمیگردونه، 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);
```

RoleProvider 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 یا ["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 امکان دیباگ از راه دور وجود ندارد، ولی گفتم شاید موردی رو از قلم انداخته باشم).

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۰۳ ۱۶:۵۳

- آیا فیلتر Authorize را با تگ authorization و location در وب کانفیگ با هم بکار برده‌اید؟ فقط از فیلتر Authorize در ASP.NET MVC استفاده کنید کافی است.
- بررسی کنید مقدار User.Identity.Name پس از لاگین چیست؟ (در یک صفحه معمولی)
- یک breakpoint داخل public override bool IsUserInRole قرار دهید و ببینید اصلا صدا زده می‌شود؟ چه پارامترهایی را دریافت می‌کند؟

نویسنده: سیروان عفیفی
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 یک صفحه معمولی مقدار صفر را برمیگرداند.
- breakpoint هم گذاشتم از صدا زده نمی‌شود.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۰۳ ۱۸:۲۷

اگر breakpoint شما صدا زده نمی‌شود، یکبار کوکی‌های برنامه را کلا پاک کنید. مجددا تست کنید. اگر IsUserInRole صدا زده نشد، یعنی مسیر قسمت type تعریف شده در role manager، قابل یافت شدن توسط برنامه نبوده.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۰۶ ۰:۴۸

یک نکته‌ی تکمیلی

می‌توان CustomRoleProvider را کلا حذف کرد و بجای آن از FormsAuthenticationTicket که نقش‌های کاربر را هم قبول می‌کند، استفاده کرد. فقط کافی است این مقدار دهی در حین لاگین کاربر انجام شود. یک مثال:

Authorizing Users with Role-Based Security

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۲/۱۱/۰۷ ۱۱:۳۷

قسمت type مشکلی نداره، breakpoint رو وقتی داخل متد GetRolesForUser قرار میدم فراخوانی میشه ولی چون مقدار Id صفر درنظر گرفته شده نمی‌تونه Role موردنظر رو برای این Id توی دیتابیس پیدا کنه، موقع لاگین فقط UserName و Password توسط سیستم Binding برای متد Login فرستاده میشه :

وقتی در فرم لاگین هم به صورت دستی Id رو ارسال میکنم، باز هم پیام Attempted to perform an unauthorized operation. دریافت میکنم. آیا تغییری دیگری در View لاگین نیاز هست اعمال بشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۰۷ ۱۴:۲۴

در حین لاگین، 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 وهله‌ای 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 ری‌دایرکت FormsAuthentication انتساب دهید.

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۲/۱۱/۰۷ ۱۵:۲۹

ممنون، 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 ویژگی Authorize و تعیین Role و یا User مزین می‌کنم میدهد خطای Attempted to perform an unauthorized operation را میدهد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۰۷ ۱۵:۴۷

Role Provider شما عمل نمی‌کند. کلا کلاس آن‌را حذف کنید و از روش Authorizing Users with Role-Based Security بعد از لاگین 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: ۱۳۹۲/۱۱/۰۷ ۲۱:۸

روش خیلی خوبیه ممنون، توی مقاله که ذکر کردیدخودش سه روش رو جهت چک کردن Role Membership پیشنهاد داده:
1- استفاده از ویژگی PrincipalPermissionAttribute :

```
[PrincipalPermissionAttribute(SecurityAction.Demand, Role = "MyRole")]
```

2- فراخوانی متد IsInRole
3- از طریق web.config
ولی می‌تونیم از فیلتر Authorize به جای مورد اول استفاده کنیم. مورد بعدی : کد زیر رو باید داخل رویداد Application_Start اضافه کرد :

```
AppDomain.CurrentDomain.SetPrincipalPolicy(PrincipalPolicy.WindowsPrincipal);
```

نویسنده: abbas
تاریخ: ۱۳۹۲/۱۱/۱۸ ۰:۴۳

سلام. با mvc 4 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 .

نویسنده: اس ام
تاریخ: ۱۳۹۲/۱۲/۲۹ ۵:۱۴

سلام من هم با MVC 5 نوشتم ولی RoleProvider صدا زده نمیشه.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۲۹ ۹:۷

در دات نت 4.5، مشکل طولانی بودن حاصل BinaryFormatter serialization برطرف شده (نزدیک 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.
- وصله‌ی نسخه‌ی ویندوز 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.

نویسنده: اس ام
تاریخ: ۱۳۹۳/۰۱/۰۱ ۳:۴۹

سلام ،سال نو مبارک ممنون از پاسخ شما.
من از vs 2013 update1 و win 8.1 استفاده میکنم و برنامه رو با .NetFrameWork 4.5.1 تست کردم.یعنی شما می‌فرمایید که اگه این بسته‌ی ارتقاع رو نصب کنم مشکل حل میشه؟ (مشکل صدا زدن RoleProvider)
اگر هم تو سیستم لوکال خودم حل شه! به سیستم عامل Host که دسترسی ندارم .بخوام اونم به روز کنم.
من در دات نت 4 که با وب فرم کار می‌کردم همیشه از RoleProvider سفارشی استفاده می‌کردم و مشکلی نداشت.با مطالعه مقاله شما علاقه مند شدم تا از این به بعد برنامه هامو با MVC توسعه بدم. ولی چون آموزش شما با MVC4 هست و الان که MVC5 عرضه شده سوالات زیادی ذهنم رو مشغول کرده یکی اینکه آیا احتمال داره طی این زمان که از انتشار این مقاله می‌گذره روش‌ها عوض شده و یا روش بهتری برای MVC ارائه شده باشه؟
بازم ممنون از زحمتتتون.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۰۱ ۹:۱۱

- تمام هاست‌ها به دلایل امنیتی، سیستم عامل و وابستگی‌های آن‌را مرتبا به روز نگه می‌دارند. این مورد، اصل اول رعایت مسایل امنیتی هست.
- MVC5 فقط یک افزونه است برای MVC4 و MVC4 هم یک افزونه است برای MVC3. در MVC5 افزونه‌ای به نام ASP.NET Identity نیز ارائه شده‌است.
- پروژه‌ی سورس باز دیگری نیز در سایت به نام Iris membership برای پوشش مسایل بحث جاری تهیه شده‌است.

نویسنده: رشوند
تاریخ: ۱۳۹۳/۰۱/۲۶ ۱۷:۱۸

سلام؛بنده هم همین مشکل رو دارم.
برای اینکه قسمت‌های مختلف که شما مطمئنا از ان اگاه نیستید و شاید بران تاثیر بگذاره رو نداشته باشید.
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 بلافاصله 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: ۱۳۹۳/۰۱/۲۶ ۱۷:۲۶

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 Authorize را به صورت Global تعریف کنید : «... امکان تعریف AuthorizeAttribute در فایل global.asax.cs و متد RegisterGlobalFilters آن به صورت سراسری نیز وجود دارد ...»

نویسنده: رشوند
تاریخ: ۱۳۹۳/۰۱/۲۷ ۱۳:۴۶

آقای نصیری سپاس.
مشکل حل شد, توضیح بدم اگر کسی مشکل من رو داشت دیگه نپرسه!
لینکی که دادید رو قبلا خوانده بودم ولی دقت نکردم در انتها ذکر کرده بودید *تعریف فیلترهای سراسری*
این کد رو متد Application_Start فایل Global.asax.cs اضافه کردم:

```
GlobalFilters.Filters.Add(new System.Web.Mvc.AuthorizeAttribute());
```

در همه جا اگر فرد لاگین کره بود می‌تونه دسترسی داشته باشه.
بعد در صفحه Home هم [AllowAnonymous] رو در ابتدا اضافه کردم تا بدون در نظر گرفتن لاگین اجازه دسترسی رو بده.

نویسنده: ایلیا اکبری فرد

تاریخ: ۱۳۹۳/۰۷/۱۷ ۱۷:۹

با سلام.

چرا مقدار returnUrl در متد login همواره نال است؟

```
[HttpPost]
[ValidateAntiForgeryToken]
[ValidateCaptchaAttribute]
[AllowAnonymous]
public virtual ActionResult Login(LoginWithCaptchaViewModel vm, string returnUrl)
{
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۱۷ ۱۸:۱۶

مقدار کوئری استرینگ returnUrl ، توسط سیستم اعتبارسنجی و redirect خودکار توسط آن بر اساس آدرس loginUrl در وب کانفیگ ، مقدار دهی می‌شود.

نویسنده: سعیدجلالی
تاریخ: ۱۳۹۳/۰۷/۲۴ ۱۰:۳۰

خوب با توجه به ارائه Identity 2 و قرارگیری آن در قالب پیش فرض MVC5 برای Visual Studio 2013 Update 2
آیا 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 بدهم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۲۴ ۱۰:۴۲

- 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: ۱۳۹۳/۰۸/۲۴ ۱۵:۱۰

با عرض سلام و خسته نباشید

میشه به یه اکشن متد گفت که فقط همان کاربری که لاگین کرده به این متد دسترسی داشته باشه مثلا به این شکل :

```
[Authorize(Users = User.Identity.Name)]
البته این کد جواب نمیدهد برای فهم کلام نوشتم
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۸/۲۴ ۱۵:۲۴

همینقدر که بنویسید [Authorize] کافی است.

عنوان: ASP.NET MVC #19
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۰۱ ۱۴:۱۶:۰۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

مروری 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 Caching سبب خواهیم شد تا اصلا کدی اجرا نشود. در ASP.NET MVC این هدف از طریق بکارگیری فیلتری به نام OutputCache 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 متد یا حتی به یک کنترلر نیز می‌توان اعمال کرد. به این ترتیب HTML نهایی حاصل از View متناظر با اکشن متد جاری فراخوانی شده، Cache خواهد شد. سپس زمانیکه درخواست بعدی به سرور ارسال می‌شود، نتیجه دریافت شده، همان اطلاعات Cache شده قبلی است و عملا در سمت سرور کدی اجرا نخواهد شد. در اینجا توسط پارامتر Duration، مدت زمان معتبر بودن کش حاصل، برحسب ثانیه مشخص می‌شود. VaryByParam مشخص می‌کند که اگر متدی پارامتری را دریافت می‌کند، آیا باید به ازای هر مقدار دریافتی، مقادیر کش شده متفاوتی ذخیره شوند یا خیر. در اینجا چون متد Index 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 نخواهند کرد و حاصل نهایی از Cache خواهنده می‌شود.
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، مثلا 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 اطلاعاتی 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 متد Index اضافه شده است. مقدار آن به ViewBag.Msg انتساب داده شده و سپس در View ، در بین تگ‌های h2 نمایش داده خواهد شد. همچنین یک فرم ساده هم جهت ارسال parameter به متد Index اضافه شده 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 TextBox نمایش داده شده یکبار مثلا 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. علت این است که مقدار VaryByParam به none تنظیم شده است و صرفنظر از ورودی کاربر، همان اطلاعات کش شده قبلی بازگشت داده خواهد شد. برای رفع این مشکل، متد Index را به نحو زیر تغییر دهید، به طوریکه مقدار VaryByParam به نام پارامتر متد جاری اشاره کند:

```
[OutputCache(Duration = 60, VaryByParam = "parameter")]
public ActionResult Index(string parameter)
```

در ادامه مجددا برنامه را اجرا کنید. اکنون یکبار مقدار Test1 را به سرور ارسال کنید. سپس مقدار Test2 را ارسال نمائید. مجددا 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 خواهند 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 Client، Server و ServerAndClient 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، مثلا محتوای لیست یک منوی پویا، محل قرارگیری اطلاعات 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) VaryByHeader : اطلاعات، بر اساس هدرهای مشخص شده، کش می‌شوند. برای مثال مرسوم است که از Accept-Language در اینجا استفاده شود تا اطلاعات مثلا فرانسوی کش شده، به کاربر آلمانی تحویل داده نشود.
د) VaryByCustom : در این حالت نام یک متد استاتیک تعریف شده در فایل global.asax.cs باید مشخص گردد. توسط این متد کلید 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 تعریفی در قسمت Remote validation بکارگرفته 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

یکی دیگر از تنظیمات ویژگی OutputCache، پارامتر CacheProfile 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 OutputCache، کل خروجی یک 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 cache substitution که به donut caching هم معروف است (چون آن‌را می‌توان به شکل یک donut تصور کرد!) در ASP.NET MVC

قابل استفاده است.

```
@{ Response.WriteSubstitution(ctx => DateTime.Now.ToShortTimeString()); }
```

همانطور که ملاحظه می‌کنید برای تعریف یک چنین اطلاعاتی باید از متد 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 partial viewها نیز مزین به OutpuCache هستند. اما صفحه اصلی درج کننده اطلاعات این سه partial view فاقد ویژگی Output کش است. در این حالت تنها اطلاعات این partial viewها کش خواهند شد و سایر قسمت‌های صفحه با هر بار درخواست از سرور، مجددا بر اساس اطلاعات جدید به روز خواهند شد. حالت توصیه شده نیز همین مورد است و متد Response.WriteSubstitution را صرفا جهت اطلاعات عمومی درنظر داشته باشید.

استفاده از امکانات Data Caching به صورت مستقیم

مطالبی 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 حین نوشتن Unit tests دردسر کمتری دارد و mocking آن ساده است؛ از این جهت که بر اساس HttpContextBase تعریف شده‌است.
در ادامه یک کلاس کمکی نوشتن اطلاعات در cache و سپس بازیابی آن‌را 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 متد، ابتدا نیاز است فضای نام این کلاس تعریف شود و سپس برای نمونه متد HttpContext.CacheInsert در دسترس خواهد بود. HttpContext یکی از خواص تعریف شده در شیء کنترلر است که با ارث بری کنترلرها از آن، همواره در دسترس می‌باشد.

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 متد CacheRead آن‌را دریافت نمود. اگر متد CacheRead نال برگرداند به معنای خالی بودن کش است. بنابراین یکبار اطلاعات را از بانک اطلاعاتی دریافت نموده و سپس آن‌را کش خواهیم کردیم.

البته هستند ORMهایی 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 قشقرقی 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: ۲۳:۰۹:۲۷ ۱۳۹۱/۰۲/۱۱

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 بدهیم؟

نویسنده: وحید نصیری
تاریخ: ۲۳:۳۶:۳۲ ۱۳۹۱/۰۲/۱۱

در حالت تئوری، بله می‌شود. مثلا پارامتر duration که بر حسب ثانیه است را مقدار دهی کرد (یک تبدیل واحد ساده است). یا 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 مشخصی اصطلاحا Recycle می‌کنه؛ یعنی برنامه ری استارت میشه و کش از دست خواهد رفت. البته این زمان در تنظیمات IIS قابل تغییر است: (^)

نویسنده: مهدی غیاثی
تاریخ: ۱:۱۶ ۱۳۹۱/۰۶/۲۷

آقای نصیری، آیا راهی هست که این caching به ازای هر کاربر انجام بشه؟ مثلا 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: ۹:۱۳ ۱۳۹۱/۰۶/۲۷

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: ۲۰:۴۶ ۱۳۹۱/۱۱/۲۲

در مورد متد CacheInsert بالا چندتا سوال داشتم:
1) ما چرا از httpcontext استفاده کردیم .
2) نحوه‌ی فراخوانی ان به چه شکل است .
با تشکر

نویسنده: وحید نصیری
تاریخ: ۲۱:۱ ۱۳۹۱/۱۱/۲۲

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 دقیقا this.HttpContext است که برای آن یک متد الحاقی به نام CacheInsert تهیه شده. بنابراین نحوه فراخوانی آن در یک اکشن متد به صورت HttpContext.CacheInsert. **this** 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 متد یک extention از نوع httpcontext هست

نویسنده: وحید
تاریخ: ۱۳۹۲/۱۰/۱۷ ۰:۵۱

با سلام اگر بخوام در یک لایه دیگر از insertCache یا readCache 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. اسمبلی 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

ممنون از شما
منظور 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 متد 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 = "*")]
```

نویسنده: 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 میخواد view2 بازگشت داده شود، اطلاعات کش بشه. من اومدم هرکدوم 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 return view 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؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۲۴ ۱۹:۸

خیر. از VaryByParam = "id" استفاده کنید (در متن فوق بیشتر در مورد آن بحث شده؛ یعنی به ازای idهای مختلف، کش‌های مختلف و متناظر خروجی نهایی متد). در این حالت نیازی به return redirect نیست که آدرس صفحه را تغییر می‌دهد. حالت return view امکان مسیردهی فایل view را هم دارد، برای انتخاب فایل‌های مختلف؛ مثلا: ("return View("~/Views/Items/Details.cshtml

نویسنده: علیرضا
تاریخ: ۱۳۹۳/۰۸/۰۹ ۱۱:۱۵

من متد تولید منو رو در یک helper نوشتم. چگونه می‌توانم آن را کش کنم؟

نویسنده: 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 مستقیما 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.

نویسنده: فرزین بیاتی
تاریخ: ۱۳۹۳/۱۲/۲۳ ۱۸:۲۲

دوستان سلام
من یک اکشن دارم که به این صورت اون رو کش کردم:

```
[OutputCache(Duration = 36000, VaryByParam = "productId;productsGalleryId", Location =
System.Web.UI.OutputCacheLocation.Server)]
        public ActionResult ShowProductPage(int productId, int productsGalleryId)
{
.
.
.
.
}
```

حالا میخوام وقتی یک محصول مشخص و یا یک گالری محصولات مشخص تغییر کرد فقط کش مربوط به اون حذف بشه ، ممنون میشم راهنمایی بفرمائید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۲/۲۳ ۱۹:۳۵

« اگر اطلاعات موجود در دیتابیس را تغییر دهیم چه کار کنیم که کشینگ رفرش شود؟ »

نویسنده: فرزین بیاتی
تاریخ: ۱۳۹۳/۱۲/۲۳ ۲۰:۵۱

ممنون از پاسختون؛ با استفاده از این روش:

```
Response.RemoveOutputCacheItem(Url.Action("ActionName", "ControllerName"));
```

تمامی کش‌های مربوط به این اکشن خاص، حذف میشن در صورتی که من میخوام به این نحو عمل کنم:
مثلا اکشن با مقدار productId=10 صدا زده می‌شه و خروجی مربوط به اون کش میشه
دفعه بعد اکشن با مقدار productId=11 فراخوانی میشه و خروجی اون هم کش میشه
حالا productId=10 ویرایش میشه و یا حذف میشه؛با روش بالا کلیه‌ی کش‌های مربوط به اکشن حذف میشن در صورتی که من میخوام فقط کش‌های مربوط به productId=10 حذف بشن، در واقع حذف هم Vary By Param بشه، ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۲/۲۳ ۲۰:۵۹

- امکان حذف کش یک پارامتر خاص هم وجود دارد:

```
HttpResponse.RemoveOutputCacheItem("/Home/ShowProductPage/param1");
```

در اینجا param1 نام یکی از پارامترهای ذکر شده‌ی در VaryByParam است و کل کش مربوط به param1 را حذف می‌کند (بدون حذف کش سایر پارامترها).
- اگر نیاز به کنترل بیشتری دارید باید سوئیچ کنید به Donut Output Caching .

عنوان: ASP.NET MVC #20
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۰۳ ۱۸:۰۹:۰۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

تهیه گزارشات تحت وب به کمک WebGrid

WebGrid از ASP.NET MVC 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 نیست و پروژه MVC Contrib یا شرکت Telerik 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 از n» 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 سایت وجود دارد. برای مثال images/id.jpg. ستونی برای نمایش تصویر متناظر با هر کارمند نیز باید اضافه شود.
14- به ازای هر کارمند، تعدادی پروژه هم وجود دارد. پروژه‌های متناظر 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 نمائید (یک کارمند که می‌تواند تعداد پروژه منتسب 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 متد index با پارامترهای sort و غیره 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 متد 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)
{
    @(((grid.PageIndex + 1) * grid.RowsPerPage) - (grid.RowsPerPage - 1));
}

@helper WebGridPageLastItem(WebGrid grid)
{
    if (grid.TotalRowCount < (grid.PageIndex + 1 * grid.RowsPerPage))
    {
      @grid.TotalRowCount;
    }
    else
    {
      @((grid.PageIndex + 1) * grid.RowsPerPage);
    }
}

<h2>Employees List</h2>

@{
    var grid = new WebGrid(
```

```
        source: Model,
        canPage: true,
        rowsPerPage: 10,
        canSort: true,
        defaultSort: "FirstName"
    );
    var salaryPageSum = 0;
    var taxPageSum = 0;
    var rowIndex = ((grid.PageIndex + 1) * grid.RowsPerPage) - (grid.RowsPerPage - 1);
}

<div id="container">
    @grid.GetHtml(
            tableStyle: "webgrid",
            headerStyle: "webgrid-header",
            footerStyle: "webgrid-footer",
            alternatingRowStyle: "webgrid-alternating-row",
            selectedRowStyle: "webgrid-selected-row",
            rowStyle: "webgrid-row-style",
            htmlAttributes: new { id = "MyGrid" },
            mode: WebGridPagerModes.All,
            columns: grid.Columns(
                 grid.Column(header: "#",
                             style: "text-align-center-col",
                             format: @<text>@(rowIndex++)</text>),
                 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"),
                 grid.Column(header: "Image",
                             style: "text-align-center-col",
                             format: @<text><img alt="@item.Id" src="@Url.Content("~/images/" +
@item.Id + ".jpg")" /></text>),
                 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");

                             }),
                 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);
                             }),
                 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"
```

```
                                        );
                                    }),
                    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"))
                         )
                     )

    <strong>Page:</strong> @(grid.PageIndex + 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 () {
        $('#MyGrid tbody:first').append(
        '<tr class="total-row"><td></td>\
         <td></td><td></td><td></td>\
         <td><strong>Total:</strong></td>\
         <td>@string.Format("${0:n0}", @salaryPageSum)</td>\
         <td>@string.Format("${0:n0}", @taxPageSum)</td>\
         <td></td><td></td><td></td></tr>');
    });
</script>
}
```

توضیحات ریز جزئیات View فوق

تعریف ابتدایی شیء WebGrid و مقدار دهی آن

در ابتدا نیاز است یک وهله 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 فعال 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 = ((grid.PageIndex + 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 خواهند 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 خواهند 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: #000;
}

.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 typed object، مقدار id گرید مشخص شده است. از این id در حین کار با jQuery استفاده خواهیم کرد.

تعیین نوع Pager

پارامتر دیگری که در متد grid.GetHtml تنظیم شده است، mode: WebGridPagerModes.All می‌باشد. WebGridPagerModes یک enum با محتوای زیر است و توسط آن می‌توان نوع Pager گرید را تعیین کرد:

```
[Flags]
public enum WebGridPagerModes
{
        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 متد کمکی grid.Columns، اینکار را انجام داد.

متد کمکی grid.Column، یک وهله از شیء WebGridColumn 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 توانمندترین پارامتر متد 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 گزارش سازی، اما اینبار با کدهای سی شارپ) و مقدار فرمت شده نهایی را به صورت یک رشته بر گردانید. این رشته نهایتا در سلول جاری درج خواهد شد.

اگر از پارامتر فرمت استفاده نشود، همان مقدار فیلد جاری بدون تغییری رندر می‌گردد.

حداقل به دو نحو می‌توان پارامتر فرمت 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);
                      }
```

مستقیما از توانمندی‌های Razor استفاده کنید. مثلا یک تگ کامل را بدون نیاز به محصور سازی آن بین "" شروع کنید. سپس item@ به وهله‌ای از رکورد در دسترس اشاره می‌کند که در اینجا وهله‌ای از شیء کارمند است.

و یا همانند روشی که برای محاسبه جمع حقوق هر صفحه مشاهده می‌کنید، مستقیما 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 حتما 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 مستقیما در کلاس کارمند تعریف نشده است. بنابراین می‌توان آن‌را با مقدار دهی صحیح پارامتر format ایجاد کرد:

```
grid.Column(header: "Image",
                  style: "text-align-center-col",
                  format: @<text><img alt="@item.Id" src="@Url.Content("~/images/" + @item.Id +
".jpg")" /></text>),
```

در این مثال، تصاویر کارمندها در پوشه images واقع در ریشه سایت، قرار دارند. به همین جهت از متد 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 وهله 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

متد Grid.GetHtml، یک رشته را بر می‌گرداند. بنابراین در هر چند سطح که نیاز باشد می‌توان یک گرید را بر اساس اطلاعات دردسترس رندر 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 متد subGrid.GetHtml دریافت و بازگشت می‌دهیم. این رشته در سلول جاری درج خواهد شد. به نوعی یک گزارش master detail یا sub report را تولید کرده‌ایم.

اضافه کردن دکمه‌های ویرایش، حذف و انتخاب

هر سه دکمه ویرایش، حذف و انتخاب در ستون‌هایی سفارشی قرار خواهند گرفت. بنابراین مقدار دهی header و format متد grid.Column کفایت 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 متد item.GetSelectLink می‌باشد. این متد جزو 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 متد کمکی به نام‌های WebGridPageFirstItem و WebGridPageLastItem تهیه شده‌اند که آن‌ها را در ابتدای View ارائه شده، مشاهده نمودید:

```
<strong>Page:</strong> @(grid.PageIndex + 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. مثلا:

```
@section script{
<script type="text/javascript">
    $(function () {
        $('#MyGrid tbody:first').append(
        '<tr class="total-row"><td></td>\
         <td></td><td></td><td></td>\
         <td><strong>Total:</strong></td>\
         <td>@string.Format("${0:n0}", @salaryPageSum)</td>\
         <td>@string.Format("${0:n0}", @taxPageSum)</td>\
         <td></td><td></td><td></td></tr>');
    });
</script>
}
```

در این مثال به کمک jQuery با توجه به اینکه id گرید ما MyGrid است، یک ردیف سفارشی که همان جمع محاسبه شده است، به tbody 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 master page سایت، استفاده نمائید:

```
<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 ۱۹:۱۸:۴۵

آفرین بر همت عالی‌تان.

نویسنده: ilia
تاریخ: ۱۳۹۱/۰۲/۰۳ ۱۹:۳۷:۵۵

تشکر فراوان آقای نصیری . خسته نباشید. خواندن مقالات شما لذت بخش و خیلی کاربردی و مفیده .

نویسنده: Mohsen Bahrzadeh
تاریخ: ۱۳۹۱/۰۲/۰۳ ۲۲:۲۲:۵۵

خیلی جالب هست استاد منتظر ادامش هستیم.

نویسنده: Naser Tahery
تاریخ: ۱۳۹۱/۰۲/۰۳ ۲۳:۲۵:۰۰

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 ۰۰:۲۲:۵۱

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 ۰۹:۴۰:۴۲

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 ۲۳:۰۲:۵۴

ممنون.واقعا روم نشد تشکر نکنم.

نویسنده: mojtaba kaviani
تاریخ: ۱۳۹۱/۰۲/۰۵ ۰۰:۵۶:۱۵

ممنون از اینکه همیشه جدیدترین تکنولوژی ها رایگان آموزش میدی منم چند تا سایت با mvc ساختم واقعا عالیه از جمله http://efish.farsedu.ir که بیش از 120000 کاربر داره ولی افت سرعتی وجود نداره

نویسنده: Atoma
تاریخ: ۱۳۹۱/۰۲/۰۵ ۱۴:۱۶:۴۰

ضمن تشکر فراوان از این همت عالی و وقتی که میذارید 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 فعلا از WebForm ها استفاده می کردید تا MVC بالغ تر شود و بعدا سراغش می رفتید؟

بازم ممنونم ازتون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۰۵ ۱۶:۴۶:۴۷

برای توضیحات بیشتر لطفا 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 بدید 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: ۱۳۹۱/۰۷/۱۶ ۱۰:۴۶

از لطفتان بسیار سپاس گذارم.
خیلی خیلی ممنون

نویسنده: رضا
تاریخ: ۱۳۹۱/۰۷/۲۳ ۱۳:۱۱

رکورد select شده با این دستور قابل ارجاع و بازیابی نیست و موقع اجرا، خطا میدهد:

```
@RenderPage("~/views/path/_partial_view.cshtml", new { Employee = grid.SelectedRow })
```

چگونه میشود رکورد انتخابی را به‌جا selectشدن به‌صورت لینکی درآورد که به صفحه جزئیات مشخصات کارمند برود. یعنی در قسمت routeValues باید چی‌رو پاس بدیم؟
ممنونم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۲۳ ۱۴:۴۶

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 })
}
```

محتویات فایل _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: ۱۳۹۱/۱۲/۰۸ ۱۶:۳۱

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: ۱۳۹۱/۱۲/۰۸ ۱۶:۴۱

- متد index یک چنین امضایی دارد

```
public ActionResult Index(string sort, string sortdir, int? page = 1)
```

زمانیکه این اطلاعات را دارید، برای مثال از LINQ استفاده کرده و با استفاده از متدهای Skip ، Take و OrderBy کار بازیابی قطعه‌ای از اطلاعات مورد نظر را انجام دهید.
- تعداد کل سطرها را هم کوئری بگیرید جداگانه و کش کنید. روش برای ارسال آن به یک View همانند کلیه روش‌های قابل استفاده در MVC است.

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۱/۱۲/۰۸ ۲۳:۲۰

متشکرم.

نویسنده: محمد رعیت پیشه
تاریخ: ۱۳۹۲/۰۸/۱۹ ۱۲:۳۷

هنگام استفاده، من این پیغام رو دریافت کردم

```
The type or namespace name 'WebGrid' could not be found.
```

با اضافه کردن رفرنس زیر به فایل Web.config مشکل حل شد.

```
<compilation debug="true" targetFramework="4.0">
      <assemblies>
        <add assembly="System.Web.Helpers, Version=1.0.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35" />
      </assemblies>
    </compilation>
```

آیا هر بار باید این رفرنس رو خودمون به صورت دستی اضافه کنیم؟

نویسنده: 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. یک سری نکات هست که باید برای ارتقاء به MVC4 مدنظر داشت: « نحوه ارتقاء برنامه‌های موجود MVC3 به MVC4 »

نویسنده: محمد رعیت پیشه
تاریخ: ۱۳۹۲/۰۸/۱۹ ۱۶:۲۰

نگارش 4.
این System.Web.Helpers, Version= **1.0.0.0** رو هم که به System.Web.Helpers, Version= **2.0.0.0** در لوکال تغییر میدم مشکلی وجود نداره، اما روی هاست اگر اضافه نکنم همون خطای قبلی رو میده و اگر هم اضافه کنم خطای زیر 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 .

نویسنده: محمد رعیت پیشه
تاریخ: ۱۳۹۲/۰۸/۱۹ ۱۶:۵۷

مشکل حل شد.
دستور Install-Package Microsoft.AspNet.WebPages رو در کنسول زدم و سپس System.Web.Helpers.dll رو از پوشه bin 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: ۱۳۹۲/۰۸/۱۹ ۱۷:۱۶

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: ۱۳۹۲/۱۰/۰۷ ۱۰:۰

چطور میشه برای قسمت هدر گرید هم یک نوع فرمت ساخت؟
مثلا می‌خواهیم در هدر یک Checkbox قرار بدیم.
با تشکر.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۰/۰۷ ۱۰:۳۳

یک مثال در این مورد WebGrid Helper with Check All Checkboxes

نویسنده: mostafa
تاریخ: ۱۳۹۲/۱۰/۰۷ ۱۰:۴۵

سلام؛ تو view که داریم میخوایم چند تا کنترل مثه گرید ویو و repeater داشته باشیم، واسه این کار چه جوری باید کنترلررو بنویسیم؟ میشه چند متد به یک view بفرستیم؟ مشکلم سر فرستادن مقدار از کنترلر به ویوئه. چون کنترلر یه خروجی بیشتر نداره.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۰/۰۷ ۱۱:۳۲

می‌تونید از ViewModel استفاده کنید.

نویسنده: طاهری
تاریخ: ۱۳۹۲/۱۱/۰۹ ۱۷:۱۰

سلام
اگه بخوام 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 شده‌است 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 متد:

```
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

بله. Inline Editing With The WebGrid

نویسنده: طاهری
تاریخ: ۱۳۹۲/۱۱/۱۲ ۱۰:۵۱

من یه 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 خواهند داشت. همچنین بعد از این تغییر، جایی که new WebGrid دارید، 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 Aتا تاریخB گزارش میخواد.دو مقدار BوA 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 »

عنوان: ASP.NET MVC #21
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۰۵ ۱۷:۴۸:۰۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

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 فرمت XML یا Json 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 وبی را که شبیه به برنامه‌های دسکتاپ 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 عموما 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، صرفا 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 ایی، استفاده از کتابخانه‌های مرتبط با jQuery می‌باشد؛ از این جهت که 100ها افزونه برای کار با آن توسط گروه وسیعی از برنامه نویس‌ها در سراسر دنیا تاکنون تهیه شده است. به علاوه فریم ورک jQuery تنها منحصر به اعمال Ajax ایی نیست و از آن جهت دستکاری DOM (document object model) و CSS صفحه نیز می‌توان استفاده کرد. همچنین حجم کمی نیز داشته، با انواع و اقسام مرورگرها سازگار است و مرتبا 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

فایل vsdoc 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 کاهش 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 jquery-xyz.js مراجعه نمائید.

محل الحاق اسکریپت‌های عمومی مورد نیاز برنامه نیز بهتر است در فایل master page یا layout برنامه باشد که به صورت پیش فرض اینکار انجام شده است.
سایر فایل‌های اسکریپتی که در این پوشه مشاهده می‌شوند، یک سری افزونه عمومی یا نوشته شده توسط تیم ASP.NET MVC برفراز jQuery هستند.

به چهار نکته نیز حین استفاده از اسکریپت‌های موجود باید دقت داشت:
الف) همیشه از متد 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 js مهم هستند. ابتدا باید کتابخانه اصلی jQuery ذکر شود و سپس افزونه‌های آن‌ها.
ج) اگر اسکریپت‌های jQuery در فایل layout سایت تعریف شده‌اند؛ نیازی به تعریف مجدد آن‌ها در Viewهای سایت نیست.
د) اگر View ایی به اسکریپت ویژه‌ای جهت اجرا نیاز دارد، بهتر است آن‌را به شکل یک section داخل view تعریف کرد و سپس به کمک متد 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 VS.NET به کمک NuGet این امکان را خواهد یافت تا شما را از آخرین به روز رسانی‌های این کتابخانه‌ها مطلع کند.

آشنایی با Ajax Helpers توکار ASP.NET MVC

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 SetViewData(ViewDataDictionary viewData);
    }
}
```

علاوه بر خاصیت Html که وهله‌ای از آن امکان دسترسی به Html helpers توکار ASP.NET MVC را در یک View فراهم می‌کند، خاصیتی به نام Ajax نیز وجود دارد که توسط آن می‌توان به تعدادی متد AjaxHelper توکار دسترسی داشت. برای مثال توسط متد Ajax.ActionLink می‌توان قسمتی از صفحه را به کمک ویژگی‌های Ajax، به روز رسانی کرد.

مثالی در مورد به روز رسانی قسمتی از صفحه به کمک متد Ajax.ActionLink

ابتدا نیاز است فایل Views\Shared_Layout.cshtml را اندکی ویرایش کرد. برای این منظور سطر الحاق jquery.unobtrusive-ajax.min.js را به فایل layout برنامه اضافه نمائید (اگر این سطر اضافه نشود، متد Ajax.ActionLink 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 متد Index کلیک راست کرده و گزینه Add view را انتخاب کنید. یک View خالی را به آن اضافه نمائید. همچنین بر روی متد EmployeeInfo کلیک راست کرده و با انتخاب گزینه Add view در صفحه ظاهر شده یک partial view را اضافه نمائید. جهت تمایز بین partial view و view هم بهتر است نام partial view با یک underline شروع شود.
کدهای partial view مورد نظر را به نحو زیر تغییر دهید:

```
@model MvcApplication18.Models.Employee

<strong>Name:</strong> @Model.Name
```

سپس کدهای View متناظر با متد 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">
    <img src="@Url.Content("~/Content/images/loading.gif")" alt="loading..."  />
</div>
```

توضیحات جزئیات کدهای فوق

متد Ajax.ActionLink لینکی را تولید می‌کند که با کلیک کاربر بر روی آن، اطلاعات اکشن متد واقع در کنترلری مشخص، به کمک ویژگی‌های jQuery Ajax دریافت شده و سپس در مقصدی که توسط UpdateTargetId مشخص می‌گردد، بر اساس مقدار InsertionMode، درج خواهد شد (می‌تواند قبل از آن درج شود یا پس از آن و یا اینکه کل محتوای مقصد را بازنویسی کند).
HttpMethod آن هم به POST تنظیم شده تا با IE مشکلی نباشد. از این جهت که IE پیغام‌های GET را کش می‌کند و مساله ساز خواهد شد. توسط پارامتر routeValues، آرگومان مورد نظر به متد EmployeeInfo ارسال خواهد شد.
به علاوه یکی دیگر از خواص کلاس AjaxOptions، برای معرفی حالت بروز خطایی در سمت سرور به نام OnFailure در نظر گرفته شده است. در اینجا می‌توان نام یک متد JavaScript ایی را مشخص کرده و پیغام خطای عمومی را در صورت فراخوانی آن به کاربر نمایش داد. یا توسط خاصیت Confirm آن می‌توان یک پیغام را پیش از ارسال اطلاعات به سرور به کاربر نمایش داد.
به این ترتیب در مثال فوق، id=1 به متد EmployeeInfo به صورت غیرهمزمان ارسال می‌گردد. 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 (InsertionMode.Replace). متد Ajax.ActionLink از این جهت داخل div تعریف شده‌است که پس از کلیک کاربر و جایگزینی محتوا، محو شود. اگر نیازی به محو آن نبود، آن‌را خارج از div تعریف 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 تعریف 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 مجددا مخفی خواهد شد.

به علاوه اگر به کدهای فوق دقت کرده باشید، از متد Request.IsAjaxRequest نیز استفاده شده است. به این ترتیب می‌توان تشخیص داد که آیا درخواست رسیده از طرف jQuery Ajax 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 متدهای 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 که توسط تیم ASP.NET MVC 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 وفق دهنده این اطلاعات با متد jQuery.Ajax از صفحه جاری، که امکان کش شدن آن‌را به سادگی میسر

می‌سازد.

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 مستقیما به کمک jQuery نیز 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.Ajax نیز می‌توان درخواست‌های Ajax ایی خود را به سرور ارسال کرد. در اینجا type نوع http verb مورد نظر را مشخص می‌کند که به POST تنظیم شده است. Url آدرس کنترلر را دریافت می‌کند. البته حین استفاده از متد توکار Ajax.ActionLink، این لینک به صورت خودکار بر اساس تعاریف مسیریابی برنامه تنظیم می‌شود. اما در صورت استفاده مستقیم از jQuery.Ajax باید دقت داشت که با تغییر تعاریف مسیریابی برنامه نیاز است تا این Url نیز به روز شود.

سه سطر بعدی نوع اطلاعاتی را که باید به سرور POST شوند مشخص می‌کند. نوع json است و همچنین contentType 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 کاراکترهای زیادی هستند که ممکن است مشکل ساز شده و باید پیش از ارسال به سرور اصطلاحا escape یا encode شوند.

متداول است از پارامتر success برای دریافت نتیجه عملیات متد jQuery.Ajax استفاده شود. اما در اینجا از پارامتر complete آن استفاده شده است. علت هم اینجا است که return PartialView یک رشته را بر می‌گرداند. 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 کرده‌ایم، 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 viewها توصیه می‌شود؛ برای مثال می‌توان اطلاعات یک مدل را به همراه یک گرید 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 متد جدید 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 متد jQuery.Ajax، مطابق کدهای فوق دسترسی یافت. اینبار چون خروجی کنترلر تعریف 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 برعکس.

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 add view را انتخاب کنید. در صفحه باز شده، گزینه create a stronlgly typed view را انتخاب کرده و قالب scaffolding را هم بر روی list قرار دهید. سپس گزینه ایجاد partial view را نیز انتخاب کنید. نام آن‌را هم _SearchEmployeeInfo وارد نمائید. برای نمونه خروجی حاصل به نحو زیر خواهد بود:

```
@model IEnumerable<MvcApplication18.Models.Employee>

<table>
    <tr>
        <th>
            Name
        </th>
    </tr>

@foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.Name)
        </td>
    </tr>
}

</table>
```

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 ایی 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 Html.BeginForm از متد Ajax.BeginForm استفاده شده است. به کمک آن اطلاعات Html.TextBox تعریف شده، به کنترلر Home و متد SearchEmployeeInfo آن، بر اساس HttpMethod تعریف 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 برنامه جهت وفق دادن این اطلاعات unobtrusive به اطلاعات مورد نیاز متد jQuery.Ajax 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>
```

کتابخانه کمکی «ASP.net MVC Awesome - jQuery Ajax Helpers»

علاوه بر متدهای توکار Ajax همراه با ASP.NET MVC، سایر علاقمندان نیز یک سری Ajax helper را بر اساس افزونه‌های jQuery تدارک دیده‌اند که از آدرس زیر قابل دریافت هستند:

http://awesome.codeplex.com/

افزودن فرم‌ها به کمک jQuery.Ajax 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، مجددا یک فرم رندر می‌شود. بنابراین اعتبار سنجی سمت کلاینت آن دیگر کار نخواهد کرد مگر اینکه با مقدار دهی خاصیت OnSuccess، مجددا متد unobtrusive.parse را فراخوانی کنیم.

نظرات خوانندگان

نویسنده: ilia
تاریخ: ۱۹:۰۹:۳۶ ۱۳۹۱/۰۲/۰۵

با سلام. مقالات شما را با جدیت دنبال می کنم . خیلی برام کاربرد دارند. متشکرم آقای نصیری .

نویسنده: amir hosein jelodari
تاریخ: ۱۹:۴۴:۰۸ ۱۳۹۱/۰۲/۰۶

سلام ... خسته نباشید ... این رو هم امتحان کنید :

/http://jsaction.codeplex.com

نویسنده: Naser Tahery
تاریخ: ۲۲:۰۴:۰۴ ۱۳۹۱/۰۲/۰۶

ممنون.
آیا استفاده از این افزونه /http://awesome.codeplex.com رایگان است؟

نویسنده: وحید نصیری
تاریخ: ۲۲:۲۶:۵۰ ۱۳۹۱/۰۲/۰۶

مجوز آن «Microsoft Public License» است. به این معنا که مجاز هستید از آن در شکل «باینری» در هر نوع پروژه‌ای استفاده کنید.

نویسنده: Naser Tahery
تاریخ: ۲۳:۰۷:۵۹ ۱۳۹۱/۰۲/۰۶

عذرخواه. ولی متوجه نمیشم در شکل باینری یعنی چی؟

نویسنده: وحید نصیری
تاریخ: ۲۳:۱۷:۲۴ ۱۳۹۱/۰۲/۰۶

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 باینری (یعنی فایل‌های dll کامپایل شده آن) در هر نوع پروژه تجاری، غیرتجاری، باز، بسته ... بدون محدودیت استفاده کنید. اما اگر سورس آن‌ها را مستقیما به پروژه خود اضافه و کامپایل کنید، نیاز است تا سورس کارتان را هم ارائه دهید.

نویسنده: علیرضا
تاریخ: ۱۶:۷ ۱۳۹۱/۰۴/۲۳

با سلام و درود و تشکر فراوان از شما
استاد من وقتی در partial View این @Molde.Name رو ست میکنم
در مرورگر **Name:** MvcApplication7.Models.Employee. Name این را مشاهده میکنم
علت چیست؟
سپاس از شما

نویسنده: وحید نصیری
تاریخ: ۱۶:۴۳ ۱۳۹۱/۰۴/۲۳

Molde نیست. Model است همچنین M آن هم باید بزرگ باشد تا با @model اشتباه نشود.
در کل امکان دیباگ پروژه‌های 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 نباشید
در روش استفاده مستقیم از ajax در jquery , خاصیت content type با DataType چه تفاوتی دارد .

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۲۳ ۱۷:۳۷

contentType معادل است با mime type 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 dataType در jQuery نوع اطلاعات برگشتی از سرور رو مشخص می‌کنه. برای مثال اگر xml ذکر شود، اطلاعات دریافتی از سرور را به فرمت xml در اختیار شما قرار می‌دهد.

نویسنده: مولانا
تاریخ: ۱۳۹۲/۰۱/۲۷ ۱۹:۴

با 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>
        آیتم‌های انتخاب شده حذف خواهند شد. آیا تأیید می‌کنید؟
    </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: ۱۳۹۲/۰۱/۲۷ ۲۰:۴۱

اینجا کار می‌کنه: (با استفاده از فایرباگ بررسی کنید آیا خطایی 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)

نویسنده: میثم خوشقدم
تاریخ: ۱۳۹۲/۰۴/۲۰ ۱۱:۲۵

سلام

خسته نباشید

در مقالتون فرمودین بجای Ajax.BeginForm می‌توان از ajax.$ استفاده نمود . اما در ستفاده از Ajax داده به صورت دستی در پارامتر Data پاس داده شده . آیا امکان این هست که Ajax 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 unobtrusive با HttpPostFileBase مشکل داره و می‌خوام از Helper دت نت استفاده نکنم و مستقیما 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 ASP.NET MVC به سرور توسط jQuery Ajax » اطلاعات یک فرم رو به صورت خودکار توسط متد form.serialize به سرور ارسال می‌کند.

- نمی‌تونید از Ajax معمولی (یا به عبارتی XMLHttpRequest) برای ارسال فایل استفاده کنید. یا باید از سیلورلایت یا فلش استفاده کنید، یا از مرورگرهایی که XMLHttpRequest *Level 2* را پشتیبانی کنند (از IE 10 به بعد مثلا)، امکان Ajax upload 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 نوشتید 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. onSuccess پس از پایان کار عملیات Ajax ایی فراخوانی می‌شود. در آنجا متد یاد شده را بر روی Id محتوای پویای بارگذاری شده فراخوانی کنید. یک نمونه مثال دیگر آن استفاده از این روش در مطلب « نمایش فرم‌های مودال Ajax ایی 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 متد jQuery.Ajax 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 حتما از Return Json استفاده کنم تا اطلاعات درخواستی نمایش داده بشه؟ آیا حتما باید از Return Json استفاده کرد ؟ و یا در کد نویسی من جایی اشکال هست ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۸/۲۵ ۱۱:۳۱

return View نهایتا 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 jQuery Ajax یک محتوای HTMLایی 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. ضمنا بهتر است از return PartialView برای این نوع کارهای خاص Ajaxایی 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 مثلا 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.MapRoute برنامه 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 ای‌جکسی 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
خیلی ممنوم ار 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

- AJAX اختراع 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 IE 8 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.Contect) <br/>
@Html.ValidationMessageFor(x=>x.Contect)
<br/>
@Html.ValidationSummary()
```

```
    <input type="submit" value="Save" />
}
</div>
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۸ ۱۴:۳۴

$.validator.unobtrusive.parse باید در جایی فراخوانی شود که کار load اولیه را انجام داده:

```
@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 ایی 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: ۱۳۹۲/۱۱/۱۹ ۲۰:۲۴

با تشکر از آموزش خوبتون،
امکان ویرایش Confirm در Ajax.BeginForm یا Ajax.ActionLink وجود داره؟
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 ؟',{prefix: 'dnt',buttons: { 'تایید': true, 'انصراف':
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: 'نظر شما ثبت شد بعد از تایید نمایش داده میشود', type:
'success', layout: 'center', timeout: 4000 });"
}))
```

منظورم نمایش پیغام به کاربر که عملیات موفق آمیز بوده ، یعنی وقتی متد OnSuccess اجرا میشود اطمینان داشته باشیم که Action مربوطه کاملا اجرا شده و اگه مشکلی در حین اجرای Action پیش بیاد این متد فراخونی نمیشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۹ ۲۱:۲۸

- فایل jquery.unobtrusive-ajax.js سورس باز هست و جزو پروژه. خاصیت confirm آن مستقیما از window.confirm استاندارد مرورگرها استفاده می‌کند و قابلیت سفارشی سازی ندارد. فقط یک متن را می‌شود اینجا تنظیم کرد.
- AjaxOptions خاصیت 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 مربوط به Ajax.BeginForm یک پارشال ویو هم در مدال رندر شود اون موقع چرا validator.unobtrusive.parse عمل نمیکند؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۸/۲۲ ۱۸:۱۹

این متد را باید در زمان نمایش نهایی فرم مودال، فراخوانی کنید. مطابق معمول روال jQuery، رخداد document ready فرم دوم (یا رخدادهایی مانند open، show یا open complete و امثال آن مختص به فرم مودال) را باید مقدار دهی کنید؛ نه فراخوانی آن متد، در فرم جاری. یک مثال

عنوان: ASP.NET MVC #22
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۰۶ ۲۲:۳۵:۰۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

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.CurrentThread.CurrentCulture

بر این اساس دات نت می‌تواند تشخیص دهد که برای مثال خروجی متد 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.CurrentThread.CurrentUICulture

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.CurrentThread.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 -> Intenet options -> Genarl 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؟ خوشبختانه 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 فایل‌هایی 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.resx را در نظر بگیرید. برای ایجاد فایل منبع اسپانیایی متناظر، باید فایلی 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.
حین تهیه برنامه‌ها نیازی نیست تا مستقیما 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 متد 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 متد Html.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 فایل cs همراه آن‌را به public تغییر دهید. پیش فرض آن internal 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 های برنامه، نخواهیم داشت.

سپس دو جفت «نام-مقدار» را در فایل resx وارد کنید. مثلا نام را Name و مقدار آن‌را «Employee Name» و سپس نام دیگر را NameIsNotRight و مقدار آن‌را «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. مثلا نام را Name و مقدار آن‌را «نام کارمند» وارد نمائید. سپس کلید یا نام NameIsNotRight و مقدار «لطفا نام را وارد نمائید» 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 هربار 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 کرده‌ایم، 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);
            }
        }
    }
}
```

بنابراین برای استفاده از آن در هر View ایی تنها کافی است بنویسیم:

```
@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 مثلا پیغام‌های اعتبار سنجی را بومی کرد یا حین استفاده از متد 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، Html.LabelFor و امثال آن به صورت خودکار مورد استفاده قرار خواهند گرفت.

نکته‌ای در مورد کش کردن اطلاعات

در این مثال اگر فیلتر OutputCache را بر روی متد Index تعریف کنیم، حتما 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 نداره؟
2- اگه بخوایم قالب بندی و direction رو هم بر اساس cultureInof تعیین کنیم، می تونیم داخل resx مثلا خاصیتی به اسم dir تعریف 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.resx باشه، فایل‌های بعدی باید xyz.lang.resx باشند.
ضمنا برای این مسایل پوشه‌ی مخصوصی به نام «App_GlobalResources» در ASP.NET وجود دارد، ولی در ASP.NET MVC توصیه نمی‌شود. چون حاصل این فایل‌ها در زمان اجرا توسط موتور ASP.NET کامپایل می‌شوند و یک dll جداگانه را درست می‌کنند که مناسب استفاده در شرایط Unit test نیست (مقادیر نال خواهند بود). بنابراین بهتر است فایل‌های منبع هر View، داخل همان پوشه مربوطه تعریف شود تا بتوان ارتباط منطقی آن‌هارا پیدا کرد و همچنین در این حالت مشکلی با آزمون‌های واحد هم نخواهد بود.
- این روش هم خوبه. چون نهایتا میشه رشته رو توسط Razor خوند و در View یا master page 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 می‌شود. یکبار تست کنید، جالب است.

نویسنده: A. Karimi
تاریخ: ۱۱:۵۳:۲۶ ۱۳۹۱/۰۲/۰۸

چون مدل‌های 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 DataAnnonation ارثبری کردم و با گرفتن رشته حاوی مشخصات Resource با استفاده از Reflection به Reousrce Type دسترسی پیدا کردم. به نظر شما راه بهتری هست؟ (با بردن Resourceها به Assemblyها دیگر هم مشکل داشتیم.)

نویسنده: وحید نصیری
تاریخ: ۱۳:۰۷:۱۳ ۱۳۹۱/۰۲/۰۸

یک پروژه نمونه که مدل آن به اسمبلی دیگری منتقل شده. همچنین منابع آن هم در اسمبلی جداگانه‌ای قرار دارند و نهایتا از تمام این‌ها در پروژه اصلی استفاده شده و بدون مشکل کار می‌کند: (^)
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 کجای کار مشکل داشت که این روش برایمان مشکل ساز شده بود! اما قریب به یقین مشکل internal نبود.

گذشته از این موضوع؛ کلاً جدا کردن Resource ها در یک Assembly دیگر بهتر است یا در پروژه UI؟ البته مطمئناً کمترین حسن آن استفاده مجدد Resourceها در پروژه‌هایی است که پلتفرمهای مختلف دارند مثل Win و Web یا حتی Mobile. به طور عملی در پروژه‌ای این جداسازی Resource را انجام ندادم. شما تجربه این کار را در پروژه‌های بزرگ داشتید؟

نویسنده: وحید نصیری
تاریخ: ۰۹:۳۵:۱۴ ۱۳۹۱/۰۲/۰۹

فکر نمی‌کنم کسی این دور و اطراف بزرگتر از SharePoint پروژه‌ای داشته باشه. SharePoint هم به وفور از فایل‌های Resx استفاده می‌کنه. محل آن‌ها هم پوشه‌های 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 ؟

ممنون

نویسنده: وحید نصیری
تاریخ: ۱۲:۲۳ ۱۳۹۱/۰۷/۲۴

فایل Index_cshtml.Designer.cs ذکر شده، یک فایل code behind متناظر 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 داده هایی هم که مدیر سایت در database وارد میکنه دارای این ویژگی باشه تا به هر زبانی که کاربر میخواد 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 خواهید 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

تاریخ: ۱۳۹۱/۱۱/۲۳ ۲۰:۳۱

شما برنامه نویس هستید. روشی خاصی نمی‌تونه شما رو محدود کنه. فقط دست آخر باید همین فریم ورک و جزئیات اون رو از صفربازنویسی 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: ۱۳۹۱/۱۱/۳۰ ۱۱:۵۲

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 اسم EntitiesResource درست کرد و پروپرتی‌ها رو 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: ۱۳۹۲/۰۲/۲۹ ۱۶:۴۰

سلام

ResourceType یکبار می‌بایست برای Display ست شود و یکبار هم برای ErrorMessageResourceType

که هر دوی این‌ها معمولا 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: ۱۳۹۲/۰۲/۲۹ ۱۷:۴۷

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 نباشید.
من دو تا فایل Resource داشتم برای زبان انگلیسی و فارسی آیا مقادیر فایل‌های Resource را می‌توان داخل کد C# مقداردهی کرد مثلا یک 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

ممنون؛ من مثلا یه ریسورس ساختم به اسم lang.resx که دیفالت هستش و یدونه هم برای فارسی به اسم lang.fa.resx یا خود مثال‌های شما. سوالم اینه که اصلا ASP.NET از کجا متوجه میشه فایل lang.fa متعلق به زبان فارسی هستش؟ استاندارد توکاری وجود داره که مثلا آخر اسم‌ها به fa یا en یا ... ختم بشه؟ همچنین گزینه Access modifier برای ریسورس‌های من کلا غیر فعال هستش.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۵ ۲۳:۵۴

- این خلاصه نام‌ها، استاندارد جهانی هستند.
- احتمالا گزینه‌ی Custom Tool فایل منبع را تغییر داده‌اید.

نویسنده: میثم فغفوری
تاریخ: ۱۳۹۲/۱۱/۱۶ ۱۳:۴۱

چند زبانه کردن کلاس‌های POCO توسط Localresourcesها 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.CurrentUICulture = 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 فیلدی خصوصیت [Requied] رو به تنهایی 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: ۱۳۹۲/۱۱/۳۰ ۲۰:۱۲

با EF 6.0.2 مشکلی مشاهده نشد. یک مثال برای آزمایش: Sample31.cs

نویسنده: بهزاد
تاریخ: ۱۳۹۲/۱۱/۳۰ ۲۰:۴۷

ممنون درست شد مشکل این بود که من با ابزار ef power tools خروجی SQL میگرفتم و بعد روی دیتابیس اجرا میکردم تا ساخته شه ظاهرا 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: ۱۳۹۳/۰۱/۲۸ ۱۵:۵۸

با سلام؛ بلافاصله بعد از استفاده (ایجاد) Index_cshtml. **fa** .resx 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: ۱۳۹۳/۰۱/۲۸ ۱۷:۴۳

فایل al.exe پوشه 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 VS.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 را برای یافتن al.exe جستجو کنید. اگر موجود است، مسیر یاد شده را دستی درست کرده و فایل al.exe و کانفیگ‌اش را در آنجا کپی کنید.
اطلاعات بیشتر

عنوان: ASP.NET MVC #23
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۰۷ ۱۱:۱۲:۰۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

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 صرفا 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، بلافاصله 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 4 تغییر دهید (برای نمونه ASP.NET MVC 3.0 مبتنی بر دات نت فریم ورک 4 است).
بعد از این تغییر، بازهم موفق به اجرای برنامه‌های ASP.NET MVC بر روی IIS 5.x نخواهیم شد؛ چون در آن زمان مفاهیم مسیریابی و Routing که اصل و پایه ASP.NET MVC هستند 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، مجددا 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 مجددا بر روی دکمه «Add» کلیک کنید.
در صفحه باز شده، مسیر Executable را C:\WINDOWS\Microsoft.NET\Framework\v4.0.30319\aspnet_isapi.dll وارد 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. مثلا 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، js 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 پسوند 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 IIS 6.x 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 IIS 6.x باید به دو نکته مهم دیگر نیز دقت داشت:

الف) ASP.NET 4 به صورت پیش فرض در IIS 6.x 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 ASP.NET 3.5 است، از آن نمی‌توانید جهت اجرای برنامه‌های ASP.NET MVC 3 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 IIS 6 اضافه نمائید و سپس در تنظیمات سایت، از این Application pool جدید استفاده نمائید.

البته روش صحیح و اصولی کار با IIS 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 نمائید.

اطلاعات تکمیلی:

نکات نصب برنامه‌های ASP.NET 4.0 بر روی IIS 6

مروری بر تاریخچه محدودیت حافظه مصرفی برنامه‌های ASP.NET در IIS

اجرای برنامه‌های ASP.NET MVC بر روی IIS 7.x ویندوز 7 و ویندوز سرور 2008

اگر برنامه ASP.NET MVC در IIS 7.x 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 IIS 5,x 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 مجددا به تنظیمات IIS اضافه خواهند شد.

نظرات خوانندگان

نویسنده: Salehi
تاریخ: ۲۱:۱۰:۰۴ ۱۳۹۱/۰۲/۰۷

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. متشکر

نویسنده: محمد شهریاری
تاریخ: ۶:۴۵ ۱۳۹۱/۰۷/۰۲

سلام
آیا MVC باید روی سرور نصب باشه ؟ یا فقط با نصب .net 4 روی سرور کفایت میکنه . مشکلی که دارم بعد از آپلود فایل‌ها روی سرور از System.web.Helper خطا میگیره و میگه نمی‌تونه این dll رو load کنه .

```
</ "add assembly="System.Web.Helpers, Version=1.0.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35>
```

نویسنده: وحید نصیری
تاریخ: ۸:۱۸ ۱۳۹۱/۰۷/۰۲

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: ۱۰:۳۳ ۱۳۹۱/۱۰/۰۸

سلام و ادب : لطفا نحوه اجرای یک وب سایت MVC را روی IIS 7 توضیح دهید. چون اگهURL Action بنویسم ، حل میشه ولی زمانیکه فقط LocalHost را می‌نویسم ارور میده و Route نمیکنه . مرسی

نویسنده: وحید نصیری
تاریخ: ۱۱:۳۵ ۱۳۹۱/۱۰/۰۸

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 IIS 7 به بعد که بر روی آن سیستم حداقل دات نت 4 نصب باشد. البته با IIS 6 هم می‌شود؛ ولی ابتدا باید تنظیمات عنوان شده در بحث را به IIS6 اعمال کرد.

در VS 2010 روی پروژه در VS.NET 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 VS 2012 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 Copy local 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 آدرسو میزنم run میشه
ولی دیفالت روت خودش عمل نمیکنه و از webconfig خطا میگیره
www.test.com>>>>>>>webconfig Erorr
www.test.com/home.index<<<<<is True
رو سرورم dotnet 4, iis7 نصب ولی هنوز دیفالت روت مشکل داره.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۲۰ ۱۸:۶

بسته‌های کامل MVC نصب هستند روی سرور؟
ELMAH را به برنامه اضافه کنید تا جزئیات خطاها را بهتر بتوانید لاگ و مشاهده کنید (مهم).
همچنین این خطاها در قسمت لاگ‌های سرور هم ثبت می‌شوند که بهتر است جزئیات آن‌ها را بررسی کنید.

نویسنده: reza110
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 مسریابی را پسونددار کنیم چگونه به روش مناسبی می‌توانیم همه جای پروژه را کنترل کنیم که مسیریابی دچار

مشکل نشود ازجمله در area
-3 چگونه بفهمیم که iis یکپارچه است یا کلاسیک
4 - آیا iis7 مد یکپارچه آن در ویندوز سرور 2003 و 2008 قابل نصب است
-5آیا برای ویندوز 8 تنظیم خاصی نیاز دارد . من یک مثال ساده را اجرا کردم و برنامه را بر روی 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 deafualt.aspx و در لود آن مستقیما 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: ۱۳۹۳/۰۸/۱۹ ۱۲:۶

- .* نیست. داخل پرانتز نوشتم. درستش *. است.
- اگر از متدهای توکار خود ASP.NET MVC برای تولید لینک‌ها استفاده کنید، این لینک‌ها صرفا 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. احتمالا روی دات نت 2 است؛ بجای 4.
توضیحات بیشتر در اینجا
- یک اکشن متد ساده برای صفحه‌ی Home طراحی کنید.

نویسنده: reza110
تاریخ: ۱۳۹۳/۰۸/۲۰ ۹:۲۸

با سلام و عرض ارادت
در iis5 مربوط به ویندوز ایکس پی هر نوع .* یا *.* یا *. وارد کنید خطا می‌گیرد.
در مورد شماره 2 چگونه می‌توان با متدهای توکار مثلا چنین ویویی 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: ۱۳۹۳/۰۸/۲۰ ۹:۴۸

- 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/....مسیر کامل فایل", 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 لاگ ویندوز ثبت می‌شوند. آن‌ها را بررسی کنید. همچنین ELMAH را هم نصب کنید تا خطاها را برای بررسی بیشتر لاگ کند.
- روش‌های قدیمی را با MVC کار نکنید. صفحه‌ی اول سایت، همان صفحه‌ای است که در مسیریابی پیش فرض تعریف شده‌است. یعنی همان اکشن متد Index در کنترلر Home، به همراه View ایی که مد نظر شما است.

عنوان: ASP.NET MVC #24
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۰۸ ۰۰:۲۶:۰۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

مروری بر نمونه سؤالات ASP.NET MVC 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 ایی از صفحه‌ای صادر شده و خروجی زیر را از اکشن متدی دریافت کند:

```
["Adventure Works","Contoso"]
```

کدام نوع خروجی اکشن متد 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 متد 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 رخ داده 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 متد 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 مدنظر داشته باشید تا برنامه به درستی کار کند؟
a) تنظیم IIS به نحوی‌که تمام درخواست‌ها را بر اساس wildcard خاصی به aspnet isapi.dll هدایت کند.

b) تنظیم IIS به نحوی‌که تمام درخواست‌ها را بر اساس wildcard خاصی به aspnet_wp.exe هدایت کند.
c) تغییر برنامه به نحوی‌که تمام درخواست‌ها را به یک HttpHandler خاص هدایت کند.
d) تغییر برنامه به نحوی‌که 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 متد 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/Shared/DisplayTemplates به پوشه Views/Player/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 متد Edit کنترلری باید 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 در span ایی با id مساوی ref، 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 قمشلویی
تاریخ: ۰۱:۰۳:۰۸ ۱۳۹۱/۰۲/۰۸

با سلام از تمامی زحماتی که طی این سری آموزش ها کشیدید متشکرم خسته نباشید خدا قوت

نویسنده: مهمان
تاریخ: ۱۵:۵۱:۲۲ ۱۳۹۱/۰۲/۰۸

با سلام و تشکر. بسیاری از مطالب به شدت حرفه ای بود و بنده 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 ASP.NET MVC و البته WebAPI همچنان ادامه داشته باشد. هرچند بصورت پراکنده

نویسنده: Shima Besharat
تاریخ: ۱۶:۱۲:۴۵ ۱۳۹۱/۰۲/۰۸

با سلام و عرض خسته نباشید خدمت جناب نصیری
نکتهٔ 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: ۱۶:۲۱:۴۳ ۱۳۹۱/۰۲/۰۸

سری «پایه» همین مباحث است. شما با این اطلاعات راحت می‌تونید یک پروژه رو جمع کنید.
ASP.NET MVC 4 نسبت به نگارش 3 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 WCF با ASP.NET MVC است به نحوی شکیل!
البته مجوز نگارش 4 بتای آن که در دسترس هست، Go live است یعنی می‌تونید 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 تکنیک‌های استفاده از jQuery UI در ASP.NET MVC یا استفاده از Kendo UI در ASP.NET MVC و موارد مشابه.

نویسنده: hammid
تاریخ: ۱۸:۴۵:۰۹ ۱۳۹۱/۰۲/۰۸

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: ۱۹:۱۴:۳۶ ۱۳۹۱/۰۲/۰۸

فایل «خلاصه‌ی وبلاگ» رو به روز کردم. شامل تمام قسمت‌ها برای مطالعه آفلاین هست.

نویسنده: Na3er Faraji
تاریخ: ۰۰:۱۳:۱۴ ۱۳۹۱/۰۲/۱۰

فقط میتونم بگم متشکرم. من فعلا قسمت دوازدهم. اما دلم نیومد اینجا ازتون تشکر نکنم. بدون اغراق بهترین مطلب آموزشی بود که در وب فارسی تا الان خونده بدوم. آقای نصیری ازتون ممنونم که بدون هیچ منتی دانش خودتون رو در اختیار من و سایرین

گذاشتید.

نویسنده: صادق
تاریخ: ۰۹:۰۱:۱۲ ۱۳۹۱/۰۲/۱۰

جددا راه نداشت که لااقل تو این پست آخریتون یه تشکر کوچک نکرده باشم.

مرسی آقای نصیری عزیز مرسی :)

نویسنده: Milad Mohseni2012
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 وقفهٔ شما عادت کرده بودیم.
لطفاً اگر امکان دارد بحث MVC را با رویکرد بکارگیری ORM ها ادامه دهید.
پیشاپیش متشکرم.

نویسنده: اردلان
تاریخ: ۱۰:۴۳ ۱۳۹۱/۰۷/۱۹

اقای نصیری بسیار از شما سپاس گذارم.
بسیار عالی بود و من واقعا با خواندن مطالب این سایت خودم را از فسیل شدن نجات دادم. خیلی خیلی متشکرم.

نویسنده: 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

نویسنده: رشوند
تاریخ: ۱۸:۱ ۱۳۹۳/۰۱/۲۸

باسلام؛ آقای نصیری سپاس , بنده چون هیچی از MVC نمی‌دونستم (نمی دونم) 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 پروژه رو جمع کرد ولی در همه آموزش تا به اینجا, برای ساده‌تر شدن از بیان دیتابیس خوداری کردید.
لطفا 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؟

نویسنده: وحید نصیری
تاریخ: ۱۸:۹ ۱۳۹۳/۰۱/۲۸

گروه EF را در سایت پیگیری کنید.
ضمنا مباحث MVC در اینجا تمام نشده. مباحث تکمیلی آن‌را در گروه MVC پیگیری کنید.

عنوان: **رفع مشکل نصب به روز رسانی ASP.NET MVC 3**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۱۰ ۱۴:۲۸:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: MVC

در مورد نحوه نصب پیشنیازهای ASP.NET MVC 3 پیشتر توضیح داده شد. یک روش دیگر هم برای اینکار مهیا است؛ اگر به مشکل برخوردید 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 اینترانت به آن اضافه شده و غیره)

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

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

ب) فایل VS10-KB*.msp یا vs10-kb*.msi 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 VS 2010 اصلی وجود دارد به آن معرفی کنید. علت هم به این بر می‌گردد که پروسه نصب VS 2010 شما از ابتدا ناقص بوده است و نیاز است یک سری فایل دیگر در ابتدا بر روی سیستم نصب گردند.

اکنون کار نصب بدون مشکل پیش خواهد رفت. لازم به ذکر است که این پیغام خطا 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 گردند (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 نصاب یک نمونه جدید را ایجاد کند.

عنوان: **سایت‌های مهمی که از ASP.NET MVC استفاده می‌کنند**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۱/۱۸ ۰۹:۴۶:۰۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

عموما استفاده وسیع از نگارش‌های مختلف ASP.NET مربوط به اینترانت‌های 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 اخیرا 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 یا ASP.NET استفاده می‌کند. اگر پسوند صفحات به asp ختم شده بود، ASP کلاسیک دهه نود است. در غیراینصورت 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 کلا تغییر داد. یعنی ضرورتی ندارد که یک سایت استفاده کننده از IIS حتما 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: ۱۳۹۱/۰۱/۱۸ ۲۱:۲۴:۰۴

تشخیص mvc بودن یه سایت برام سوال بود واقعا. متشکر

نویسنده: Hooshmand Meysam
تاریخ: ۱۳۹۱/۰۱/۱۸ ۲۳:۵۰:۱۲

سلام
به نظرم، راه فنی تری را می توان انتخاب کرد؟؟ مثلا 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: ۱۳۹۱/۰۱/۱۹ ۰۰:۴۳:۳۶

الگو که همان view state خودش یک الگو است. از این افزونه استفاده کنید تا حجم view state را هم مشاهده کنید: (^) .
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: ۱۳۹۱/۰۱/۱۹ ۰۱:۱۰:۱۷

در مورد SharePoint 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: ۱۳۹۱/۰۱/۱۹ ۰۹:۱۰:۲۹

بعید می‌دونم. علتش به توسعه پذیری SharePoint بر می‌گرده 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 واقعا عظیمی است. خیلی هزینه‌بر است تبدیل آن.
برای مثال شاید همین سوال در مورد IE هم باشد. چرا IE رو با دات نت نمی‌نویسند؟ علتش این است که بعد از این همه سال میلیون‌ها دلار خرج code base آن شده. دور ریختن و دل کندن از آن واقعا سخت است.

نویسنده: شاهین کیاست
تاریخ: ۱۳۹۱/۰۵/۰۵ ۴:۴

به این لیست قطعا باید سایت جاری هم اضافه کرد.

نویسنده: احمدعلی شفیعی
تاریخ: ۱۳۹۱/۰۵/۱۳ ۳:۶

شما یه چیزی رو در نظر نگرفتید برای تشخیص. اینکه فرد از تکنولوژی غیر از دات‌نت استفاده کنه. مثلا بعضی از سایت‌ها نه ViewState داره و نه index.php معلومه. ولی php هست و نه MVC دارن!

نویسنده: مهدی غیاثی
تاریخ: ۱۳۹۱/۰۵/۱۳ ۳:۹

در تکمیل حرف دوستم احمدعلی، البته وبلاگ 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: ۱۳۹۱/۰۵/۱۳ ۸:۱۸

افزونه server spy معرفی شده سرور سایت شما رو cloudflare-nginx نشون می‌ده. nginx 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 **90** درصد موارد». به این معنا که در 10 درصد باقیمانده موارد، شخص می‌تواند از الگوهای متداول دیگری جهت شناسایی ساختار یک برنامه استفاده کند.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۳ ۸:۲۴

مآخذ:
- دقیقه 52 این ویدیو از زبان یکی از مدیران مایکروسافت.
- به علاوه به علت استفاده از دات نت در بینگ، تیم دات نت دست به بهبود گسترده‌ای 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 پلوغ 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 Bing.com با ASP.NET MVC 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. ویکی پدیا به همین نحو مدیریت می‌شود.
- ماخذ این گفته، اسکات گاتری هست (دقیقه 52 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 منصوب شده.
- این ماخذ 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 ایی 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 ۰:۱

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 __EVENTVALIDTION و __VIEWSTATE *با مقدار خالی* ، وجود خواهند داشت و حذف نمی‌شوند.

عنوان: **نحوه ارتقاء برنامه‌های موجود MVC3 به MVC4**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۰۳ ۱۰:۴۸
آدرس: www.dotnettips.info
گروه‌ها: Entity framework, MVC, ASP.Net MVC, MS Ajax Minifier

در ادامه، مراحل ارتقاء پروژه‌های قدیمی MVC3 را به ساختار جدید پروژه‌های MVC4 مرور خواهیم کرد .

1) نصب پیشنیاز

الف) نصب VS 2012

و یا

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

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

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

پس از نصب پیشنیازها

2) نیاز است نوع پروژه ارتقاء یابد

به پوشه پروژه MVC3 خود مراجعه کرده و تمام فایل‌های csproj و web.config موجود را با یک ادیتور متنی باز کنید (از خود ویژوال استودیو استفاده نکنید، زیرا نیاز است محتوای فایل‌های پروژه نیز دستی ویرایش شوند).

در فایل‌های csproj (یا همان فایل پروژه؛ که vbproj هم می‌تواند باشد) عبارت

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

را جستجو کرده و با

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

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

3) به روز رسانی شماره نگارش‌های قدیمی

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

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

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

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

این کارها را با replace in all open documents توسط notepad plus-plus به سادگی می‌توان انجام داد.

4) به روز رسانی مسیرهای قدیمی

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

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

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

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

5) به روز رسانی قسمت appSettings فایل‌های کانفیگ

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

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

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

6) رسیدگی به وضعیت اسمبلی‌های شرکت‌های ثالث

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

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

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

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

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

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

Unobtrusive.Ajax.2

Unobtrusive.Validation.2

Web.Optimization.1.0.0

و ....

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

8) اضافه کردن یک فضای نام جدید

بسته Web Optimization را از طریق NuGet دریافت کنید (برای یافتن آن bundling را جستجو کنید؛ نام کامل آن Microsoft ASP.NET Web Optimization Framework 1.0.0 است). این مورد به همراه پوشه MVC4 نیست و باید از طریق NuGet دریافت و نصب شود. (البته پروژه‌های جدید MVC4 شامل این مورد هستند)

در فایل وب کانفیگ، فضای نام System.Web.Optimization را نیز اضافه نمائید:

```
<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 EF 5 جای یک سری از کلاس‌ها تغییر کرده. مثلا ویژگی‌های ForeignKey، ComplexType و ... به فضای نام System.ComponentModel.DataAnnotations.Schema منتقل شده‌اند. در همین حد تغییر جهت کامپایل مجدد کدها کفایت می‌کند.

همچنین فایل‌های پروژه موجود را باز کرده و EntityFramework, Version=4. **1**. 0.0 را جستجو کنید. نگارش جدید 4. **4**. 0.0 است که باید اصلاح شود (این موارد را بهتر است توسط یک ادیتور معمولی خارج از VS.NET ویرایش کنید).

در زمان نگارش این مطلب EF Mini Profiler با EF 5 سازگار نیست. بنابراین اگر از آن استفاده می‌کنید نیاز است غیرفعالش کنید.

اولین استفاده از امکانات جدید MVC4:

استفاده از امکانات System.Web.Optimization که ذکر گردید، می‌تواند اولین تغییر مفید محسوب شود.

برای اینکه با نحوه کار آن بهتر آشنا شوید، یک پروژه جدید MVC4 را در VS.NET (از نوع basic) آغاز کنید. به صورت خودکار یک پوشه جدید را به نام App_Start به ریشه پروژه اضافه می‌کند. داخل آن فایل مثال BundleConfig قرار دارد. این کلاس در فایل global.asax برنامه نیز ثبت شده‌است. باید دقت داشت در حالت دیباگ (compilation debug=true در وب کانفیگ) تغییر خاصی را ملاحظه نخواهید کرد.

تمام این‌ها خوب؛ اما من به نحو زیر از این امکان جدید استفاده می‌کنم:

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

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

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

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

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

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

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

            BundleTable.Bundles.Add(newBundle);
        }

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

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

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

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

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

چند نکته مهم در این کلاس وجود دارد:

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

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

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

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

```
@using Common.WebToolkit

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

```
rel="stylesheet" type="text/css" media="print"/>

@BundleConfig.AddScripts("~/Scripts/js",
                            "~/Scripts/jquery-1.8.0.min.js",
                            "~/Scripts/jquery.unobtrusive-ajax.min.js",
                            "~/Scripts/jquery.validate.min.js")

@BundleConfig.AddStyles("~/Content/css",
                            "~/Content/Site.css",
                            "~/Content/buttons.css")
```

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

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

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

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

این مورد خصوصا در مسیریابی تصاویر مرتبط با اسکریپت‌ها و شیوه نامه‌ها مؤثر است؛ وگرنه این تصاویر تعریف شده در فایل‌های CSS یافت نخواهند شد (تمام مثال‌های موجود در وب با این مساله مشکل دارند و فرض آن‌ها بر این است که کلیه فایل‌های خود را در یک پوشه، بدون هیچگونه زیرپوشه‌ای تعریف کرده‌اید).
پارامترهای بعدی، محل قرارگیری اسکریپت‌ها و CSSهای برنامه هستند و همانطور که عنوان شد اینبار با خیال راحت می‌توانید ترتیب معرفی خاصی را مدنظر داشته باشید؛ زیرا توسط AsIsBundleOrderer به صورت پیش فرض لحاظ خواهد شد.

نظرات خوانندگان

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

سلام ... من این کارارو 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 vs2010 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 ۱۳:۴۳

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

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

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

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

- 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 ۹:۲۳

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

تاریخ: ۱۳۹۲/۰۵/۱۱ ۲۲:۱۴

دقیقا تفاوت BundleConfig.AddStyleUrl و BundleConfig.AddStyles چیه؟

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

با سلام
زمانی که در PartialViewها نیاز به Jquery Validator است در صورتی که آنها را در فایل Layout و در قسمت Body به صورت

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

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

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

در PartialViewها تعریف نماییم؟ اگه خیر پس چرا جواب نمیده؟

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

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: ۱۳۹۲/۱۲/۲۸ ۰:۳۳

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

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

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

بله. FileInfo به BundleFile تبدیل شده.

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

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

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

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

مشکل از کجا میتونه باشه ؟ ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۲۰ ۱۱:۵۰

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

نویسنده: محسن موسوی
تاریخ: ۱۳۹۳/۰۷/۱۲ ۱۰:۵

میتونه به دلیل این باشه که فایل‌های js مربوطه رو پیدا نکرده باشه.
جهت اطلاعات بیشتر
Unable to generate 'VersionQueryString' when using bundle

نویسنده: حسن اسلامی
تاریخ: 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 & Minification عمل minify کردن رو انجام بده؟
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.

نویسنده: خالقی
تاریخ: ۱۳۹۳/۱۱/۲۷ ۱۲:۹

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

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۱/۲۷ ۱۲:۱۸

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

عنوان: **تغییرات بوجود آمده در Bundling and Minification -MVC4**
نویسنده: شهروز جعفری
تاریخ: ۱۳۹۱/۰۴/۰۶ ۰:۲
آدرس: www.dotnettips.info
گروه‌ها: MVC

همانطور که در پست قبلی اشاره کردم نسخه جدید MVC با تغییرات جهت بهبود کاربری همراه بوده است.
در این پست قصد دارم شما رو با ویژگی جدید Bundling and Minification بر روی فایلهای Css و JavaScript آشنا کنم.اگر یک پروژه جدید به وسیله MVC 4 ایجاد 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 به Layout.cshtml 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=ji3nO1pdg6VLv3CVUWntxgZNf1z" rel="stylesheet" type="text/css" />
<link href="/Content/themes/base/css?v=UM624qf1uFt8dYti" rel="stylesheet" type="text/css" />
<script src="/Scripts/js?v=4h5lPNUsLiFoa0vqrItjS2Jp"></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 extension method 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 CssMinify وJsMinify 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.cshtm 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=uBlJsuIAGAF93nUTTez8" rel="stylesheet">
<script src="/bootstrap/js?v=O6HaHC8QqtB"></script>
```

عدم پشتیبانی از CDN میتونه تنها ایرادش باشه.

البت شاید من نتونستم راهی براش 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: ۱۳۹۱/۰۴/۰۶ ۸:۱۲

سلام دوست عزیز

به نظر من یکی از ایرادهای این فشرده سازی عدم استفاده expiretion و عدم اغمال تنظیمات http response header هست که 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 سایتهایی با تعداد درخواست بالا میتونه بار سنگینی به سرور تحمیل کنه

موفق باشید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۰۶ ۸:۲۲

اینطور نیست. مطابق مستندات رسمی آن ، HTTP Expires Header به مدت یکسال بر روی منابع تولیدی تنظیم می‌شود (و مرورگر مرتبا 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 ?css وجود دارد. این v به معنای نگارش است. اگر این نگارش که به صورت خودکار بر اساس هش فایل‌ها تولید می‌شود تغییر کند، مرورگر کش خود را به روز خواهد کرد.
ضمن اینکه در اینجا قرار نیست فایلی یا 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: ۱۳۹۱/۰۴/۰۶ ۸:۵۷

ممنونم 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: ۱۳۹۱/۰۴/۰۶ ۹:۳

- با توجه به محتوای فایل هست و این محاسبات فقط بر اساس درخواست مرورگر انجام می‌شود. یعنی هربار به ازای نمایش صفحات این کارها و محاسبات صورت نمی‌گیرد و اگر کش شده باشد (که شده برای یکسال)، اصلا فشاری به سرور وارد نمی‌شود چون مرورگر 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: ۱۳۹۱/۰۴/۰۶ ۹:۱۸

این مقاله مربوط به ASP.NET MVC 4 نسخه‌ی Beta هست. در نسخه‌ی RC، تغییراتی در نحوه‌ی استفاده از Bundles و Minification صورت گرفته. مثلاً متد ذیل منسوخ شده.

```
BundleTable.Bundles.RegisterTemplateBundles();
```

و متد ذیل به جای اون قرار گرفته:

```
BundleConfig.RegisterBundles(BundleTable.Bundles);
```

یا به جای دستور System.Web.Optimization.BundleTable.Bundles.ResolveBundleUrl، از Styles.Render و Scripts.Render استفاده میشه.

نویسنده: شهروز جعفری
تاریخ: ۱۳۹۱/۰۴/۰۶ ۱۰:۵۴

حق با آقای راد هستش برای اضافه کردن اسکریپت میشه 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 4 3 تا ویژگی داره:

اگر فایل FileX.min.js و همچنین فایل FileX.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.

نویسنده: امیرحسین جلوداری
تاریخ: ۱۳۹۱/۰۴/۰۶ ۲۰:۳۳

اگه اینایی که فرمودینو ، تو مقاله‌ی اصلیم اعمال کنین به نظرم خیلی بهتره !

با تشکر ...

نویسنده: میهمان
تاریخ: ۱۳۹۱/۱۲/۱۱ ۳:۳۵

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 ؟
ممنون میشوم اگر راهنمایی بفرمائید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۱۱ ۹:۴۱

من برای کارهای خودم از روش مطرح شده در مطلب ارتقاء به MVC4 استفاده می‌کنم.

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۱۱/۲۶ ۱۸:۲۴

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 بندلینگ اطلاع دارم)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۲۶ ۱۹:۱

گوگل محتوای فایل‌های جاوا اسکریپت لینک شده به صفحه را نمی‌خواند. اینجا آزمایش شده

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۱۱/۲۷ ۰:۲۸

بله حق با شماست محتویات نمی‌خونه اما این لینک‌های و آدرس‌ها تاثیر منفی میذاره مثلا 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

What does this mean?

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=qr&chld=L|0&chl=http://www.[illegible]/
- http://www.[illegible].ir/content/css?v=vQ9qEVDZjy2UDhi2HYM9W6-uFjcmcJBBMKGTP2wVcNo1
- http://www.[illegible]ir/scripts/js?v=KRSqz5-4zabGlxQ2tOcx3uLrTHd6sJue5xmXaq1sOnQ1

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۲۷ ۰:۳۴

در مورد سرعت صفحه عنوان کرده و گفته ممکن است URLهای 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.

Console | HTML | CSS | Script | DOM | Net | Cookies | YSlow | Changes | Pixel P... | FirePath | Firefin... | Refere... | CSS Us...

Home | Grade | Components | Statistics | Tools — Rulesets YSlow(V2) — Edit

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

عنوان: **تغییرات بوجود آمده در Mobile Features-MVC4**
نویسنده: شهروز جعفری
تاریخ: ۱۳۹۱/۰۴/۰۶ ۱:۱۵
آدرس: www.dotnettips.info
برچسب‌ها: MVC

یکی دیگه از امکاناتی که به MVC4 اضافه شده و برام 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 بسازیم MVC 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, جاوا اسکریپ یا CSS نیست بلکه با پشتیانی کامل از صفحات لمسی ,تبلت‌ها , Smart Phoneها ویژگی قدرتمندی را به برنامه نویس میدهد.

در ادامه قصد دارم شما را با یک صفحه ساده ساخته شده توسط این کتابخانه قذرتمند آشنا کنم.

ابتدا یک پروژه خالی MVC4 ایجاد کنید.(هدف ما بیشتر برای آشنایی با کتابخانه JQuery Mobile است پس میتوان از یک صفحه Html 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, جاوا اسکریپ و CSS هستید.

نکته ای که باید توجه کرد این است که اگر از MVC4 استفاده میکنید این فایلها چون در پوشه Content ودر Root این پوشه ایجاد میشود امکان دارد ظاهر اصلی سایت را بهم بزند و شاید هم بعضی از فایلهای جاوا اسکریپت شما اجرا نشود و این به علت ویژگی Bundling است که کل فایل هایی که در Root فولدر Content , Script قرار دارد را Bundle میکند وامکان تداخل در فایلهای CSS و جاوااسکریپت وجود دراد.که میتوان فایلهای مربوط به JQuery Mobile را در فولدرهای جداگانه نگهداری کرد.(بازهم میگم ممکن است)

نکته دیگر این است که شما زمانی که به وسیله تبلت یا مویایل خود سایت را مشاهده میکنید ممکن است سایت را خیلی ریز ببینید 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-sclae=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="conent">
            <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>
```

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 در باره این تنظیم به این سایت JQueryMobile مراجع نمایید.

```
"data-role="listview
```

همانطور که از اسمش پیداست برای مشخص کردن listview است.وباقی کد نیز مشخص است.

```
"data-filter="true
```

توسط ویژگی بالا یک فیلترینگ زیبا بر روی آیتم های listview خواهیم داشت.

```
" data-inset="true"
```

واگر مقدار true باشد لبه‌های listview به صورت گرد در خواهند آمد.

در قسمت‌های بعدی توضیحات کاملتری ارائه خواهم داد.
در ضمن اگر قلت املاعی دارم به بزرگی خودتون ببخشید.;)

عنوان: تغییرات بوجود آمده در Single Page Application (SPA)-MVC4
نویسنده: شهروز جعفری
تاریخ: ۱۳۹۱/۰۴/۰۶ ۲:۳۷
آدرس: www.dotnettips.info
برچسب‌ها: MVC

تو سری پست‌های قبلی که براتون گذاشتم به تغییرات و ویژگی‌های بوجود آمده در MVC4 اشاره کردم یکی دیگه از این ویژگیها (Single Page Application (SPA 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 بخاطر آماده نشدنش بهمراه نسخه نهاییه یا مثل خیلی از تکنولوژیها وسط راه پشیمون شده.
SPA شامل سه چیز کلی میباشد.

چند کتابخانه جاواسکریپتی برای تعامل هر چه بهتر با کلاینت به وسیله کش کردن داده‌ها upshot.JS, knockout and nav.js.
کامپنتهای افزوده شده Web API برای پشتبانی از اگوی واحد کار Unit of Work
و اسفاده از scaffolding برای سریعتر کردن کار

برای نصب کافیه که به سایت ASP.net برید یا از طریق Nuget اقدام کنید.

تصویر بالا نشان دهنده ساختار Single Page Application است.

JavaScript Libraries
شامل کتابخانه‌های معروف مثل History, Knockout و کتابخانه نسبتاً جدید Upshot است که باهم کار نمایش و اصلاح داده‌ها را در مجموعه ای از صفحات به صورت local وremote انجام میدهند.

DataController on the Server
در سمت سرور شامل کلاسی به نام DataController که مشتق شده از ApiController که عملیات insert, update ,delete را به وسیله الگوی واحد کار (Unit of Work)انجام میدهد و همچنین از validations به صورت اتوماتیک پشیبانی 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 میتونم 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 میخواید برید سراغش قبلش باید اشراف کامل روی جاوا اسکریپ و Knockout داشته باشید.

نظرات خوانندگان

نویسنده: رضا.ب
تاریخ: ۱۳۹۱/۰۴/۰۶ ۲۱:۲۴

دات نت رو جایی دست گذاشته که خیلی میتونه موفق باشه. چون تعامل راحت‌تر و کاهش تراکنش‌های HTTP هر دو باعث کاربردپذیری بالاتر میشن. و این دقیقا هدفیه که دنبال میکنه.
اگر مقدور است توضیح دهید چرا به راحتی میگین " مثل خیلی از تکنولوژیها وسط راه پشیمون شده. " ؟ و اینکه مثلا وب‌سایت‌های که همین مفهوم را پیاده‌سازی کردن - مثل توئیتر، فیس‌بوک و... - با چه تکنیکی پیاده کردن؟
یه سوال مرتبط: موتورهای جستجو چجوری در این‌جور 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 Nav.JS یه URL فرندلی میشه و کاربر حس نمیکند که فقط در "یک صفحه" است؟
ممنونم.

نویسنده: شهروز جعفری
تاریخ: ۱۳۹۱/۰۴/۰۶ ۲۲:۱۵

طبق تحقیقاتی که انجام دادم اصلا Seo friendly نیست دلیلشم بر میگرده به نحوه محاسبه Page rank که همون اسمش مشخصه به محتواتی صفحه کار داره.

نویسنده: رضا.ب
تاریخ: ۱۳۹۱/۰۴/۰۷ ۲:۵۷

اگه واقعا اینطوری که شما میگین باشه، وارد شدن در این راه به‌شدت اشتباهه. چون مسئله‌ی کاملا جدی هست. به‌نظرم میرسه، اطلاعات کمی موجوده و شاید نظر دادن در موردش درست نباشه.

نویسنده: رضا
تاریخ: ۱۳۹۱/۰۴/۰۷ ۷:۵۶

این موارد برای نوشتن برنامه‌های (سازمانی) تحت وب است که seo friendly بودن اصلا مهم نیست

نویسنده: رضا.ب
تاریخ: ۱۳۹۱/۰۴/۰۷ ۱۰:۵۳

خب با این حال، مثلا فیس‌بوک یا توئیتر چگونه SEO فرندلی هستند؟ البته اگر آنها رو جزو SPA بیاریمش!

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۰۷ ۱۱:۳۳

برنامه‌های وبی که برای سازمان‌ها نوشته می‌شوند از همان ابتدای امر با مباحث authorization و authtication محافظت خواهند شد (تمام صفحات آن‌ها). به همین جهت در این نوع برنامه‌ها به صورت خودکار دسترسی به هر نوع bot ایی بسته می‌شود که باید هم به همین ترتیب باشد.

نویسنده: شهروز جعفری
تاریخ: ۱۳۹۱/۰۴/۰۷ ۱۱:۵۵

نه اینطور نیست شما در داخل صفحاتی که موتورهای جستجو به اون دسترسی ندارند مثلاً صفحات ویرایش حذف و... میتونید استفاده کنید

نویسنده: شهروز جعفری

تاریخ: ۱۳۹۱/۰۴/۰۷ ۱۲:۱

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 همئن تکنولوژیه.

در کل توئیتر و فیس‌بوک تو کل صفحاتشون از SPA استفاده نمیکنن.

عنوان: **تغییرات بوجود آمده در Razor -MVC4**
نویسنده: شهروز جعفری
تاریخ: ۱۳۹۱/۰۴/۰۲ ۱۴:۱۵
آدرس: www.dotnettips.info
برچسب‌ها: MVC, Razor

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 method s

لیست فوق شامل برترین ویژگی‌های این نسخه است که در پستهای آینده هر کدام از این‌ها بررسی خواهند 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.

شرط ها(Conditions)

در نسخه قبلی برای استفاده از attribute ها که ممکن بود Null باشند 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 ما null باشد به صورت اتوماتیک تشخیص داده میشود و کد زیر رندر میشود

```
<div>Content</div>
```

در ضمناً کد بالا فقط مربوط به چک کردن Nullable 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 میشه گفت MVC4 تغییراتش نسبت به نسخه قبلی تو خیلی از زمینه‌ها مربوط میشه بهبود ابزارهای موجود در کل کار با این ایزار بسیار برای من لذت بخشه.
ادامه دارد...

عنوان: Globalization در ASP.NET MVC
نویسنده: یوسف نژاد
تاریخ: ۱۳۹۱/۱۰/۲۱ ۱۶:۲۵
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net MVC, .NET, Localization, Globalization, Internationalization, Resource, .resx, MVC

اگر بازار هدف یک محصول شامل چندین کشور، منطقه یا زبان مختلف باشد، طراحی و پیاده سازی آن برای پشتیبانی از ویژگی‌های چندزبانه یک فاکتور مهم به حساب می‌آید. یکی از بهترین روشهای پیاده سازی این ویژگی در دات نت استفاده از فایلهای Resource است. درواقع هدف اصلی استفاده از فایلهای Resource نیز Globalization است. Globalization برابر است با Internationalization + Localization که به اختصار به آن g11n 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 بومی‌سازی به شخصی‌سازی یک برنامه برای یک فرهنگ یا زبان خاص گفته میشود. (اطلاعات بیشتر در اینجا).
استفاده از این فایلها محدود به پیاده سازی ویژگی چندزبانه نیست. شما میتوانید از این فایلها برای نگهداری تمام رشته‌های موردنیاز خود استفاده کنید. نکته دیگری که باید بدان اشاره کرد این است که تقریبا تمامی منابع مورد استفاده در یک محصول را میتوان درون این فایلها ذخیره کرد. این منابع در حالت کلی شامل موارد زیر است:
- انواع رشته‌های مورد استفاده در برنامه چون لیبل‌ها و پیغام‌ها و یا مسیرها (مثلا 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
- و ...
نحوه بهره برداری از فایلهای Resource در دات نت، پیاده سازی نسبتا آسانی را در اختیار برنامه نویس قرار میدهد. برای استفاده از این فایلها نیز روشهای متنوعی وجود دارد که در مطلب جاری به چگونگی استفاده از آنها در پروژه‌های ASP.NET MVC پرداخته میشود.

Globalization در دات نت

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 دوحرفی 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 دوحرفی 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 استثنا هم وجود دارد (اطلاعات کامل کلیه زبانها: National Language Support (NLS) API Reference). یک فرهنگ خنثی (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 استثنا وجود دارد (^).
در دات نت دو نوع culture وجود دارد: **Culture** و **UICulture** . هر دوی این مقادیر در هر Thread مقداری منحصربه فرد دارند. مقدار Culture بر روی توابع وابسته به فرهنگ (مثل فرمت رشته‌های 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 داده‌ها استفاده کند. درواقع در دات نت با استفاده از پراپرتی‌های 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.CurrentUICulture = new CultureInfo("fa-IR");
Thread.CurrentThread.CurrentUICulture = CultureInfo.CreateSpecificCulture("fa-IR");
```

دراینجا 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 خواهم 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_GlobalResources یا App_LocalResources 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.) مثلا 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 متناظر 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 ببینید. البته ابزار دیگری برای تولید این کلاسها نیز وجود دارد. این ابزارهای توکار برای سطوح دسترسی مخلف 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 دقیقا در پراپرتی Custom Tool تایپ کنید.

Solution Explorer
Properties
References
Controls
Global
Configs.fa.resx
Configs.resx
Exceptions.fa.resx
Exceptions.resx
Paths.fa.resx
Paths.resx
Texts.fa.resx
Texts.resx
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
Custom Tool Namespace
File Name
Texts.resx
Full Path
Build Action
How the file relates to the build and deployment processes.

نکته: درباره پراپرتی مهم Build Action این فایلها در مطالب بعدی بیشتر بحث میشود.
برای تعیین سطح دسترسی Resource موردنظر به روشی 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 همنام 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. دقت کنید اگر نام فایل را در همان پنجره افزودن فایل وارد کنید ویژوال استودیو این همنامی را به صورت هوشمند تشخیص داده و تغییراتی را در پراپرتی‌های پیش فرض فایل Resource ایجاد میکند.
نکته: این هوشمندی مرتبه نسبتا بالایی دارد. بدین صورت که تنها درصورتیکه عبارت بعد از نام فایل اصلی Resource (رشته بعد از نقطه مثلا fa در اینجا) متعلق به یک کالچر معتبر باشد این تغییرات اعمال خواهد شد.
مهمترین این تغییرات این است که ابزاری را برای پراپرتی Custom Tool این فایلها انتخاب نمیکند! اگر به پراپرتی 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 Access Modifier 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: DisplayName و Required. 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 به نام Display (در فایل System.ComponentModel.DataAnnotations.dll) وجود دارد که این امکان را به صورت توکار دارد. اما قابلیت استفاده از این خاصیت تنها در MVC 3 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 کاملا به صورت دستی کلید موردنظر را از Resource معرفی شده بدست آورید. مثلا 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 موردبحث پشتیبانی میکنند.

نکته: به کار گیری این روش ممکن است در پروژه‌های بزرگ کمی گیج کننده و دردسرساز 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. معمولا برای اینکار از کوکی‌های دارای تاریخ انقضای طولانی استفاده میشود. میتوان از تنظیمات موجود در فایل کانفیگ برای ذخیره زبان پیش فرض سیستم نیز استفاه کرد.
روشی که معمولا 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.CurrentUICulture = 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.CurrentUICulture = 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 رخدادن رویدادهای 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 معمولا 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 (مثلا در فایل _Layout.cshtml) برای تعیین و تغییر زبان استفاه 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+|\s+$/g, "");
      if (x == name) {
        return unescape(y);
      }
    }
  }
</script>
```

متدهای setCookie و getCookie جاوا اسکریپتی در کد بالا از اینجا گرفته شده اند البته پس از کمی تغییر.

نکته : مطلب Cookieها بحثی نسبتا مفصل است که در جای خودش باید به صورت کامل آورده 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 دردسرساز 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 Fiddler آن:

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 و مقدار پیش فرض آن 1 است. هرچه مقدار این پارامتر بیشتر باشد زبان مربوطه اولویت

بالاتری دارد. مثلا عبارت زیر را درنظر 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 برابر 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 استفاه کرد. مثلا در یک اکشن فیلتر:

```
var langs = filterContext.HttpContext.Request.UserLanguages;
```

پراپرتی UserLanguages از کلاس Request حاوی آرایه‌ای از استرینگ است. این آرایه درواقع از Split کردن مقدار Accept-Languages 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 عناوین و نیز موقعیت فیزیکی این عناوین در مسیر صفحات درخواستی کاملا به سلیقه برنامه نویس و یا کارفرما بستگی دارد. روش کلی بهره برداری از این روش در تمام موارد تقریبا 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
);
```

دقت 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.CurrentUICulture = 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 دقیقتری بر کلیه مسیرهای موجود داشت!

استفاده از ویوهای جداگانه برای زبانهای مختلف

برای اینکار ابتدا ساختار مناسبی را برای نگهداری از ویوهای مختلف خود درنظر 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 فایلهای ویو در یک فولدر زیاد خواهد شد. روش دیگری که برای نگهداری این ویوها میتوان به کار برد استفاده از فولدرهای جداگانه با عناوین 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:

استفاه از هرکدام از این روشها کاملا 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 متد 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.CurrentUICulture.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 قبلا نیز شرح داده شد، چون متد ExecuteCore قبل از OnActionExecuted صدا زده میشود بنابراین از تنظیم درست مقدار کالچر در ثرد جاری اطمینان داریم.

روش دیگری که برای مدیریت انتخاب ویوهای مناسب استفاده از یک ویوانجین شخصی سازی شده است. مثلا 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 نسبتا 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 ساختار داده‌ها یا تعیین نوع داده ای اطلاعات موجود در یک فایل XML به کار میرود. درواقع 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 فایلهاست (تگهای resheader و data). همانطور در بالا مشاهده میکنید تگهای reheader شامل تنظیمات مربوط به فایل resx. با ساختاری ساده به صورت name/value است. یکی از این تنظیمات resmimetype فایل resource را معرفی میکند که درواقع 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 تعریف شده برای استفاده از این فایلهاست. 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 resource 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 نیز 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 فایلها (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.

- application/x-microsoft.net.object.bytearray.base64: آبجکت ابتدا باید با استفاده از یک System.ComponentModel.TypeConverter به آرایه ای از بایت سریالایز شده و سپس با فرمت base64 به یک رشته انکد شود.

نکته : امکان جاسازی کردن (embed) فایلهای resx. در یک اسمبلی یا کامپایل مستقیم آن به یک سَتِلایت اسمبلی (ترجمه مناسبی برای satellite assembly پیدا نکردم، چیزی شبیه به اسمبلی قمری یا وابسته و از این قبیل ...) وجود ندارد. ابتدا باید این فایلهای resx. به فایلهای resources. تبدیل شوند. اینکار با استفاده از ابزار Resource File Generator (نام فایل اجرایی آن resgen.exe است) انجام میشود (^ و ^). سپس میتوان با استفاده از Assembly Linker ستلایت اسمبلی مربوطه را تولید کرد (^). کل این عملیات در ویژوال استودیو با استفاده از ابزار msbuild به صورت خودکار انجام میشود!

نحوه یافتن کلیدهای Resource در بین فایلهای مختلف Resx توسط پرووایدر پیش فرض در دات نت

عملیات ابتدا با بررسی 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. مثلا 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.

Globalization در کلاینت (javascript gl1n)

یکی دیگر از موارد استفاده gl1n در برنامه نویسی سمت کلاینت است. با وجود استفاده گسترده از جاوا اسکریپت در برنامه نویسی سمت کلاینت در وب اپلیکیشنها، متاسفانه تا همین اواخر عملا ابزار یا کتابخانه مناسبی برای مدیریت gl1n در این زمینه وجود نداشته است. یکی از اولین کتابخانه‌های تولید شده در این زمینه کتابخانه jQuery Globalization است که توسط مایکروسافت توسعه داده شده است (برای آشنایی بیشتر با این کتابخانه به ^ و ^ مراجعه کنید). این کتابخانه بعدا تغییر نام داده و اکنون با عنوان Globalize شناخته میشود. Globalize یک کتابخانه کاملا مستقل است که وابستگی به هیچ کتابخانه دیگر ندارد (یعنی برای استفاده از آن نیازی به jQuery 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 gl1n 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 فایلها 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: ۱۳۹۱/۱۰/۲۱ ۲۳:۵

سلام آقای یوسف نژاد
من بعد از تلاش‌های زیاد توی پروژه‌های مختلف این مطالبی که شما نوشته اید رو پیاده سازی کردم ، ولی خیلی پراکنده.
ولی حالا می‌بینم شما به زیبایی این مطالب رو کنار هم قرار دادید.
می خواستم بابت مطلب خوب و مفیدتون و همچنین وقتی که گذاشتید تشکر کنم.
ممنونم بابت زحمات شما

اگر ممکنه برچسب MVC رو هم به مطلبتون اضافه کنید.

نویسنده: یوسف نژاد
تاریخ: ۱۳۹۱/۱۰/۲۱ ۲۳:۲۴

با سلام و تشکر بابت 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: ۱۳۹۱/۱۰/۲۲ ۱:۳

کاملا مشخصه که مطلب از روی تجربه‌ی کاریه و بسیار عالی جمع آوری شده ... ممنون ... به طرز عجیبی منتظر قسمت بعدم :دی

نویسنده: پندار
تاریخ: ۱۳۹۱/۱۲/۰۸ ۲۱:۳۸

گویا در MVC 4 این روش پاسخ نمیدهد. لطفا در این مورد برای MVC 4 راه حلی بدهید

نویسنده: محسن
تاریخ: ۱۳۹۱/۱۲/۰۸ ۲۳:۶

MVC 4 فقط یک سری افزونه بیشتر از MVC3 داره. مثلا razor آن بهبود پیدا کرده، فشرده سازی فایل‌های CSS به اون اضافه شده یا Web API رو به صورت یکپارچه داره. از لحاظ کار با فایل‌های منبع فرقی نکرده.

نویسنده: پندار
تاریخ: ۱۳۹۱/۱۲/۰۹ ۹:۲۱

متن نشانی زیر را مطالعه کنید

http://geekswithblogs.net/shaunxu/archive/2012/09/04/localization-in-asp.net-mvc-ndash-upgraded.aspx

نویسنده: محسن
تاریخ: ۱۳۹۱/۱۲/۰۹ ۹:۴۳

مطلبی 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 باشه، اصول فوق فرقی نکرده.

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۱/۱۲/۱۴ ۱۶:۵

مطلب خیلی خوبی بود کلی استفاده کردیم.
مهندس کالچر زبان کردی چی میشه؟ توی لیست منابعی که دادین گیر نیاوردم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۱۴ ۱۷:۲۲

kur هست مطابق استاندارد .

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۱/۱۲/۱۷ ۲:۱۱

سلام
اما مهندس کلاس Culture Info این مقدار قبول نمی‌کنه

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۱۷ ۹:۳

می‌تونید کلاس فرهنگ سفارشی را ایجاد و استفاده کنید.

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۱/۱۲/۱۷ ۱۰:۱۲

اما روش گفته شده نیاز به دسترسی مدیریت دارد که روی سرورهای اشتراکی ممکن نیست

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۱۷ ۱۱:۱۹

نحوه توسعه اکثر برنامه‌ها و کتابخانه‌ها در طول زمان، بر اساس تقاضا و پیگیری مصرف کننده است. اگر بعد از بیش از 10 سال، چنین فرهنگی اضافه نشده یعنی درخواستی نداشته. مراجعه کنید به محل پیگیری این نوع مسایل .

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۱/۱۲/۱۹ ۱۰:۱۴

سلام مهندس یوسف نژاد (ابتدا ممنونم از پست خوب شما)
با پیروی از پست شما
ابتدا فایل‌های ریسورس در پروژه جاری فولدر 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 فایل‌های 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 خطا صادر میشه:

```
<asp:Literal ID="Literal1"  runat="server" Text='<%$ ResourcesManager.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 'ResourcesManager.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 ست نکردم ؟

نویسنده: یوسف نژاد
تاریخ: ۱۳۹۲/۰۱/۳۱ ۱۲:۴۴

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 وارده برای این خواص را در سمت سرور انجام میدهد. کلاس پیش فرض برای اینکار در ASP.NET Web Form که از پیشوند Resources استفاده میکند تنها برای Resourceهای محلی (Local) موجود در فولدرهای پیش فرض (App_GlobalResources و App_LocalResources) 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: ۱۳۹۲/۰۱/۳۱ ۱۲:۵۲

اگر مشکلی در پیاده سازی روش بالا دارین، تمام مراحلی که من طی کردم دقیقا اینجا میارم:
ابتدا کلاس ExpressionBuilder رو 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 Attribute کلاس ResourceExpressionBuilder اهمیت چندانی ندارد! امیدوارم مشکلتون حل بشه.

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۰۲/۰۱ ۲:۲۹

ممنونم از پاسخ شما
همون روش شمارو دنبال کردم پاسخ گرفتم، اشکال از خودم بود
با تشکر از شما

نویسنده: صادق نجاتی
تاریخ: ۱۳۹۲/۱۲/۲۷ ۱۲:۰

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

عنوان: Globalization در ASP.NET MVC - قسمت دوم
نویسنده: یوسف نژاد
تاریخ: ۱۳۹۲/۰۱/۳۱ ۱۳:۵
آدرس: www.dotnettips.info
برچسب‌ها: MVC, ASP.Net MVC, Globalization, Resource, Updateable Resource

به‌روزرسانی 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 اسمبلی‌های 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 اسمبلی نهایی (و در حالت کلی نحوه استفاده از هر فایلی در اسمبلی 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 نسبتا من‌درآوردی 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 فولدر App_Data است. زیرا محتویات این فولدر توسط سیستم FCN (همان *File Change Notification*) در ASP.NET رصد نمیشود.
نکته: علت این حساسیت این است که FCN در ASP.NET تقریبا تمام محتویات فولدر سایت در سرور (فولدر App_Data یکی از معدود 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. ازآنجا که محتوای این فایلها به صورت XML ذخیره میشود بنابراین براحتی میتوان با امکانات موجود این 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 قبلی 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. اسمبلی مربوط به هر زبان هم در فولدری با عنوان زبان مربوطه ذخیره میشود. مسیر و نام فایل این اسمبلی‌ها مثلا 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 Resource Generator و Assembly 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 Enumator 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. مثلا 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. ازآنجاکه 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 متد برای 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

متاسفانه در این کلاس امکان افزودن یا ویرایش یک کلید به تنهایی 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);
    }
  }
}
```

در متد فوق از همان کلاس ResXResourceEntry که در قسمت قبل معرفی شد، استفاده شده است. از متد زیر نیز میتوان برای حالت کلی حذف یا ویرایش استفاده 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 بالا نشان داده شده‌اند استفاده شده است.

حذف یک کلید در فایل resx.

برای اینکار میتوان از متد 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

قبل از بکارگیری روشهای معرفی شده در این مطلب بهتر است ابتدا یکسری قرارداد بصورت زیر تعریف شوند:

- طبق راهنماییهای موجود در قسمت قبل یک پروژه جداگانه با عنوان Resources برای نگهداری فایلهای resx. ایجاد شود.
- همواره آخرین نسخه از محتویات 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 فولدری با عنوان Defaults در مسیر App_Data\Resources برای فراهم کردن امکان "بازگرداندن به تنظیمات اولیه" وجود داشته باشد.

برای فراهم کردن این موارد بهترین راه حل استفاده از تنظیمات Post-build event command line است. اطلاعات بیشتر درباره Build Eventها در اینجا .

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 فولدرها 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. (مربوط به TFS) و نیز خود فایل excludes.txt از لیست کپی دستور xcopy حذف میشوند و بنابراین کپی نمیشوند. درصورت 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 ازپیش تعریف شده (Macro) وجود دارند که از برخی از آنها در دستوارت فوق استفاده شده است:

$(ProjectDir) : مسیر کامل و مطلق پروژه جاری به همراه یک کاراکتر \ در انتها

$(SolutionDir) : مسیر کامل و مطلق سولوشن به همراه یک کاراکتر \ در انتها

$(OutDir) : مسیر نسبی فولدر Output پروژه جاری به همراه یک کاراکتر \ در انتها

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 موردنیاز 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.CurrentUICulture.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 ResXResourceFileManager مدیریت فایلهای این کتابخانه‌ها صورت میپذیرد:

```
public class ResXResourceFileManager
{
  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 فایلهای Resource و کلاسهای متناظر با آنهاست که براساس استانداردها باید برطبق مسیر فیزیکی آنها در پروژه باشد مثل Global.Texts یا Views.LogOn. همچنین در متد RestoreResourceFiles نمونه هایی از مسیرهای این فایلها 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 فولدر bin برنامه کپی شده سپس فولدرهای تولیدشده توسط msbuild، حذف شوند. این کار در متد CopyDlls از کلاسResXResourceFileManager 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 بلافاصله پس از کپی فایلها در مقصد با توجه به توضیحات ابتدای این مطلب سایت Restart خواهد شد که یکی از ضعفهای عمده این روش به شمار میرود.

سایر متدهای موجود نیز برای برگرداندن تنظیمات اولیه بکار میروند. در این متدها از محتویات فولدر Defaults استفاده میشود.

نکته : درصورت ساخت دوباره اسمبلی و یا بازگرداندن اسمبلی‌های اولیه، از آنجاکه وب‌سایت Restart خواهد شد، بنابراین بهتر است تا صفحه جاری بلافاصله پس از اتمام عملیات،دوباره بارگذاری شود. مثلا 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

نویسنده: بهمن خلفی
تاریخ: ۱۳۹۲/۰۲/۰۱ ۹:۲۲

با سلام خدمت شما
مطلب بسیار مفیدی است و امیدوارم ادامه دهید...

عنوان: Globalization در ASP.NET MVC - قسمت سوم
نویسنده: یوسف نژاد
تاریخ: ۱۳۹۲/۰۲/۱۸ ۰:۵۰
آدرس: www.dotnettips.info
گروه‌ها: Globalization, Resource, .resx, Global Resource, Local Resource

قبل از ادامه، بهتر است یک مقدمه کوتاه درباره انواع منابع موجود در ASP.NET ارائه شود تا درک مطالب بعدی آسانتر شود.

نکات اولیه

- یک فایل Resource درواقع یک فایل XML شامل رشته هایی برای ذخیره سازی مقادیر (منابع) موردنیاز است. مثلا رشته هایی برای ترجمه به زبانهای دیگر، یا مسیرهایی برای یافتن تصاویر یا فایلها و ... . پسوند این فایلها resx. است (مثل MyResource.resx).
- این فایلها برای ذخیره منابع از جفت داده‌های کلید-مقدار (key-value pair) استفاده می‌کنند. هر کلید معرف یک ورودی مجزاست. نام این کلیدها حساس به حروف بزرگ و کوچک **نیست (Not Case-Sensitive)** .
- برای هر زبان (مثل fa برای فارسی) یا کالچر موردنظر (مثل fa-IR برای فارسی ایرانی) می‌توان یک فایل Resource جداگانه تولید کرد. عناون زبان یا کالچر باید جزئی از نام فایل Resource مربوطه باشد (مثل MyResource.fa.resx یا MyResource.fa-IR.resx). هر منبع باید دارای یک فایل اصلی (پیش‌فرض) Resource باشد. این فایل، فایلی است که برای حالت پیش‌فرض برنامه (بدون کالچر) تهیه شده است و در عنوان آن از نام زبان یا کالچری استفاده نشده است (مثل MyResource.resx). برای اطلاعات بیشتر به قسمت اول این سری مراجعه کنید.
- تمامی فایل‌های Resource باید دارای کلیدهای یکسان با فایل اصلی Resource باشند. البته لزومی ندارد که این فایل‌ها حاوی تمامی کلیدهای منبع پیش‌فرض باشند. درصورت عدم وجود کلیدی در یک فایل Resource عملیات پیش فرض موجود در دات نت با استفاده از فرایند مشهور به fallback مقدار کلید موردنظر را از نزدیکترین و مناسبترین فایل موجود انتخاب می‌کند (درباره این رفتار در قسمت اول توضیحاتی ارائه شده است).
- در زمان اجرا موتور پیش فرض مدیریت منابع دات نت با توجه به کالچر UI در ثرد جاری اقدام به انتخاب مقدار مناسب برای کلیدهای درخواستی (به همراه فرایند fallback) می‌کند. فرایند نسبتا پیچیده fallback در اینجا شرح داده شده است.

منابع Global و Local

در ASP.NET دو نوع کلی Resource وجود دارد که هر کدام برای موقعیت‌های خاصی مورد استفاده قرار می‌گیرند:

- Resourceهای **Global** : منابعی کلی هستند که در تمام برنامه در دسترسند. این فایل‌ها در مسیر رزرو شده **APP_GlobalResources** در ریشه سایت قرار می‌گیرند. محتوای هر فایل resx. موجود در این فولدر دارای دسترسی کلی خواهد بود.

- Resourceهای **Local** : این منابع همان‌طور که از نامشان پیداست محلی! هستند و درواقع مخصوص همان مسیری هستند که در آن تعبیه شده اند! در استفاده از منابع محلی به ازای هر صفحه وب (aspx. یا master.) یا هر یوزرکنترل (ascx.) یک فایل resx. تولید می‌شود که تنها در همان صفحه یا یوزرکنترل در دسترس است. این فایل‌ها درون فولدر رزرو شده **APP_LocalResources** در مسیرهای موردنظر قرار می‌گیرند. درواقع در هر مسیری که نیاز به این نوع از منابع باشد، باید فولدری با عنوان App_LocalResources ایجاد شود و فایلهای resx. مرتبط با صفحه‌ها یا یوزرکنترل‌های آن مسیر در این فولدر مخصوص قرار گیرد.
در تصویر زیر چگونگی افزودن این فولدرهای مخصوص به پروژه وب اپلیکیشن نشان داده شده است:

نکته: دقت کنید که تنها یک فولدر App_GlobalResources به هر پروژه می‌توان افزود. همچنین در ریشه هر مسیر موجود در پروژه تنها می‌توان یک فولدر Appp_LocalResources داشت. پس از افزودن هر یک از این فولدرهای مخصوص، منوی فوق به صورت زیر در خواهد آمد:

نکته: البته با تغییر نام یک فولدر معمولی به این نام‌های رزرو شده نتیجه یکسانی بدست خواهد آمد.

نکته: در زمان اجرا، عملیات استخراج داده‌های موجود در این نوع منابع، به صورت **خودکار** توسط ASP.NET انجام می‌شود. این داده‌ها پس از استخراج در حافظه سرور کش خواهند شد.

برای روشن‌تر شدن مطالب اشاره شده در بالا به تصویر فرضی! زیر توجه کنید (اسمبلی‌های تولید شده برای منابع کلی و محلی فرضی است):

در تصویر بالا محل قرارگیری انواع مختلف فایلهای Resource و نیز محل نهایی فرضی اسمبلی‌های ستلایت تولید شده، برای حداقل یک زبان غیر از زبان پیش فرض برنامه، نشان داده شده است.

نکته: نحوه برخورد با این نوع از فایل‌های Resource در پروژه‌های Web Site و Web Application کمی باهم فرق می‌کند. موارد اشاره شده در این مطلب بیشتر درباره Web Applicationها صدق می‌کند.

برای آشنایی بیشتر بهتر است یک برنامه **وب اپلیکیشن** جدید ایجاد کرده و همانند تصویر زیر یکسری فایل Resource به فولدرهای اشاره شده در بالا اضافه کنید:

همانطور که مشاهده می‌کنید به صورت پیش‌فرض برای منابع کلی یک فایل cs. تولید می‌شود. اما اثری از این فایل برای منابع محلی نیست.

حال اگر پنجره پراپرتی فایل منبع کلی را باز نمایید با چیزی شبیه به تصویر زیر مواجه خواهید شد:

می‌بینید که خاصیت Build Action آن به Content مقداردهی شده است. این مقدار موجب می‌شود تا این فایل به همین صورت و در همین مسیر مستقیما در پابلیش نهایی برنامه ظاهر شود. در قسمت قبل به خاصیت Buil Action و مقادیر مختلف آن اشاره شده است.

هم‌چنین می‌بینید که مقدار پراپرتی Custom Tool به **GlobalResourceProxyGenerator** تنظیم شده است. این ابزار مخصوص تولید کلاس مربوط به منابع کلی در ویژوال استودیو است. با استفاده از این ابزار فایل Resource1.Designer.cs که در تصویر قبلی نیز نشان داده شده، تولید می‌شود.

حالا پنجره پراپرتی‌های منبع محلی را باز کنید:

می‌بینید که همانند منبع کلی خاصیت Build Action آن به Content تنظیم شده است. همچنین مقداری برای پراپرتی Custom Tool تنظیم نشده است. این مقدار پیش فرض را تغییر ندهید، چون با تنظیم مقداری برای آن چیز مفیدی عایدتان نمی‌شود!

نکته: برای به روز رسانی مقادیر کلیدهای منابعی که با توجه به توضیحات بالا به همراه برنامه به صورت فایلهای resx. پابلیش می‌شوند، کافی است تا محتوای فایلهای resx. مربوطه با استفاده از یک ابزار (همانند نمونه ای که در قسمت قبل شرح داده شد) تغییر داده شوند. بقیه عملیات توسط ASP.NET انجام خواهد شد. اما با تغییر محتوای این فایلهای resx. با توجه به رفتار FCN در ASP.NET (که در قسمت قبل نیز توضیح داده شد) سایت Restart خواهد شد. البته این روش تنها برای منابع کلی و محلی درون مسیرهای مخصوص اشاره شده کار خواهد کرد.

استفاده از منابع Local و Global

پس از تولید فایل‌های Resource، می‌توان از آن‌ها در صفحات وب استفاده کرد. معمولا از این نوع منابع برای مقداردهی پراپرتی کنترل‌ها در صفحات وب استفاده می‌شود. برای استفاده از کلیدهای منابع محلی می‌توان از روشی همانند زیر بهره برد:

```
<asp:Label ID="lblLocal" runat="server" meta:resourcekey="lblLocalResources" ></asp:Label>
```

اما برای منابع کلی تنها می‌توان از روش زیر استفاده کرد (یعنی برای منابع محلی نیز می‌توان از این روش استفاده کرد):

```
<asp:Label ID="lblGlobal" runat="server" Text="<%$ Resources:CommonTerms, HelloText %>" ></asp:Label>
```

به این عبارات که با فوت پررنگ مشخص شده اند اصطلاحا «عبارات بومی‌سازی» (Localization Expression) می‌گویند. در ادامه این سری مطالب با نحوه تعریف نمونه‌های سفارشی آن آشنا خواهیم شد.
به نمونه اول که برای منابع محلی استفاده می‌شود نوع ضمنی (Implicit Localization Expression) می‌گویند. زیرا نیازی نیست تا محل کلید موردنظر صراحتا ذکر شود!
به نمونه دوم که برای منابع کلی استفاه می‌شود نوع صریح (Explicit Localization Expression) می‌گویند. زیرا برای یافتن کلید موردنظر باید آدرس دقیق آن ذکر شود!

بومی سازی ضمنی (Implicit Localization) با منابع محلی عنوان کلید مربوطه در این نوع عبارات همانطور که در بالا نشان داده شده است، با استفاده از پراپرتی مخصوص **meta:resoursekey** مشخص می‌شود. در استفاده از منابع محلی تنها یک نام برای کل خواص کنترل مربوطه در صفحات وب کفایت می‌کند. زیرا عنوان کلیدهای این منبع باید از طرح زیر پیروی کند:

ResourceKey.Property

ResourceKey.Property.SubProperty یا ResourceKey.Property-SubProperty

برای مثال در لیبل بالا که نام کلید Resource آن به lblLocalResources تنظیم شده است، اگر نام صفحه وب مربوطه page1.aspx باشد، برای تنظیم خواص آن در فایل page1.aspx.resx مربوطه باید از کلیدهایی با عناوینی مثل عنوان‌های زیر استفاه کرد:

lblLocalResources.Text

lblLocalResources.BackColor

برای نمونه به تصاویر زیر دقت کنید:

```
<asp:TextBox ID="txtTest" runat="server" meta:resourcekey="txtTest" />
```

بومی سازی صریح (Explicit Localization)

در استفاده از این نوع عبارات، پراپرتی مربوطه و نام فایل منبع صراحتا در تگ کنترل مربوطه آورده می‌شود. بنابراین برای هر خاصیتی که می‌خواهیم مقدار آن از منبعی خاص گرفته شود باید از عبارتی با طرح زیر استفاده کنیم:

```
<% Resources: Class, ResourceKey $%>
```

در این عبارت، رشته Resources **پیشوند (Prefix)** نام دارد و مشخص کننده استفاده از نوع صریح عبارات بومی سازی است. Class نام کلاس مربوط به فایل منبع بوده و اختیاری است که تنها برای منابع کلی باید آورده شود. ResourceKey نیز کلید مربوطه را در فایل منبع مشخص می‌کند.

برای نمونه به تصاویر زیر دقت کنید:

```
<asp:Label ID="Label1" runat="server" Text="<%$ Resources: Resource1, String1 %
```

نکته: استفاده همزمان از این دو نوع عبارت بومی سازی در یک کنترل مجاز نیست!

نکته: به دلیل تولید کلاسی مخصوص منابع کلی (با توجه به توضیحات ابتدای این مطلب راجع به پراپرتی Custom Tool)، امکان استفاده مستقیم از آن درون کد نیز وجود دارد. این کلاسها که به صورت خودکار تولید می‌شوند، به صورت مستقیم از کلاس ResourceManager برای یافتن کلیدهای منابع استفاده می‌کنند. اما روش مستقیمی برای استفاده از کلیدهای منابع محلی درون کد وجود ندارد.

نکته: درون کلاس System.Web.UI.TemplateControl و نیز کلاس HttpContext دو متد با نامهای GetGlobalResourceObject و GetLocalResourceObject وجود دارد که برای یافتن کلیدهای منابع به صورت غیرمستقیم استفاده می‌شوند. مقدار برگشتی این دو متد از نوع object است. این دو متد به صورت مستقیم از کلاس ResourceManager استفاده **نمیکنند** ! هم‌چنین ازآنجاکه کلاس Page از کلاس TemplateControl مشتق شده است، بنابراین این دو متد در صفحات وب در دسترس هستند.

دسترسی با برنامه نویسی

همانطور که در بالا اشاره شد امکان دستیابی به کلیدهای منابع محلی و کلی ازطریق دو متد GetGlobalResourceObject و GetLocalResourceObject نیز امکان پذیر است. این دو متد با فراخوانی ResourceProviderFactory جاری سعی در یافتن مقادیر کلیدهای درخواستی در منابع موجود می‌کنند. درباره این فرایند در مطالب بعدی به صورت مفصل بحث خواهد شد.

کلاس TemplateControl

این دو متد در کلاس TemplateControl از نوع Instance (غیر استاتیک) هستند. امضای (Signature) این دو متد در این کلاس به صورت زیر است:

متد GetLocalResourceObject:

```
protected object GetLocalResourceObject(string resourceKey)
protected object GetLocalResourceObject(string resourceKey, Type objType, string propName)
```

در متد اول، پارامتر resourceKey در متد GetLocalResourceObject معرف کلید منبع مربوطه در فایل منبع محلی متناظر با صفحه جاری است. مثلا lblLocalResources.Text. ازآنجاکه به صورت پیش‌فرض موقعیت فایل منبع محلی مرتبط با صفحات وب مشخص است بنابراین تنها ارائه کلید مربوطه برای یافتن مقدار آن کافی است. مثال:

```
txtTest.Text = GetLocalResourceObject("txtTest.Text") as string;
```

متد دوم برای استخراج کلیدهای منبع محلی با مشخص کردن نوع داده محتوا (معمولا برای داده‌های غیر رشته‌ای) و پراپرتی موردنظر به کار می‌رود. در این متد پارامتر objType برای معرفی نوع داده متناظر با داده موجود در کلید resourceKey استفاده می‌شود. از پارامتر propName نیز همانطور که از نامش پیداست برای مشخص کردن پراپرتی موردنظر از این نوع داده معرفی شده استفاده می‌شود.

متد GetGlobalResourceObject:

```
protected object GetGlobalResourceObject(string className, string resourceKey)
protected object GetGlobalResourceObject(string className, string resourceKey, Type objType, string
propName)
```

در این دو متد، پارامتر className مشخص کننده نام کلاس متناظر با فایل منبع اصلی (فایل منبع اصلی که کلاس مربوطه با نام آن ساخته می‌شود) است. سایر پارامترها همانند دو متد قبلی است. مثال:

```
TextBox1.Text = GetGlobalResourceObject("Resource1", "String1") as string;
```

کلاس HttpContext

در این کلاس دو متد موردبحث از نوع استاتیک و به صورت زیر تعریف شده‌اند:

متد GetLocalResourceObject:

```
public static object GetLocalResourceObject(string virtualPath, string resourceKey)
public static object GetLocalResourceObject(string virtualPath, string resourceKey, CultureInfo
culture)
```

در این دو متد، پارامتر virtualPath مشخص کننده مسیر نسبی صفحه وب متناظر با فایل منبع محلی موردنظر است، مثل "~/Default.aspx". پارامتر resourceKey نیز کلید منبع را تعیین می‌کند و پارامتر culture نیز به کالچر موردنظر اشاره دارد. مثال:

```
txtTest.Text = HttpContext.GetLocalResourceObject("~/Default.aspx", "txtTest.Text") as string;
```

متد GetGlobalResourceObject:

```
public static object GetGlobalResourceObject(string classKey, string resourceKey)
public static object GetGlobalResourceObject(string classKey, string resourceKey, CultureInfo culture)
```

در این دو متد، پارامتر className مشخص کننده نام کلاس متناظر با فایل منبع اصلی (فایل منبع بدون نام زبان که کلاس مربوطه با نام آن ساخته می‌شود) است. سایر پارامترها همانند دو متد قبلی است. مثال:

```
TextBox1.Text = HttpContext.GetGlobalResourceObject("Resource1", "String1") as string;
```

نکته: بدیهی است که در MVC تنها می‌توان از متدهای کلاس HttpContext استفاده کرد.

روش دیگری که تنها برای منابع کلی در دسترس است، استفاده مستقیم از کلاسی است که به صورت خودکار توسط ابزارهای Visual Studio برای فایل منبع اصلی تولید می‌شود. نمونه‌ای از این کلاس را که برای یک فایل Resource1.resx (که تنها یک ورودی با نام String1 دارد) در پوشه App_GlobalResources تولید شده است، در زیر مشاهده می‌کنید:

```
//------------------------------------------------------------------------------
// <auto-generated>
//     This code was generated by a tool.
//     Runtime Version:4.0.30319.17626
//
//     Changes to this file may cause incorrect behavior and will be lost if
//     the code is regenerated.
// </auto-generated>
//------------------------------------------------------------------------------

namespace Resources {
    using System;

    /// <summary>
    ///   A strongly-typed resource class, for looking up localized strings, etc.
    /// </summary>
    // This class was auto-generated by the StronglyTypedResourceBuilder
    // class via a tool like ResGen or Visual Studio.
    // To add or remove a member, edit your .ResX file then rerun ResGen
    // with the /str option or rebuild the Visual Studio project.

[global::System.CodeDom.Compiler.GeneratedCodeAttribute("Microsoft.VisualStudio.Web.Application.Strongl
yTypedResourceProxyBuilder", "10.0.0.0")]
    [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
    [global::System.Runtime.CompilerServices.CompilerGeneratedAttribute()]
    internal class Resource1 {

        private static global::System.Resources.ResourceManager resourceMan;

        private static global::System.Globalization.CultureInfo resourceCulture;

        [global::System.Diagnostics.CodeAnalysis.SuppressMessageAttribute("Microsoft.Performance",
"CA1811:AvoidUncalledPrivateCode")]
        internal Resource1() {
        }

        /// <summary>
        ///   Returns the cached ResourceManager instance used by this class.
        /// </summary>

[global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.EditorBrowsableSt
ate.Advanced)]
        internal static global::System.Resources.ResourceManager ResourceManager {
            get {
                if (object.ReferenceEquals(resourceMan, null)) {
                    global::System.Resources.ResourceManager temp = new
global::System.Resources.ResourceManager("Resources.Resource1",
global::System.Reflection.Assembly.Load("App_GlobalResources"));
                    resourceMan = temp;
                }
                return resourceMan;
            }
        }

        /// <summary>
        ///   Overrides the current thread's CurrentUICulture property for all
        ///   resource lookups using this strongly typed resource class.
        /// </summary>

[global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.EditorBrowsableSt
ate.Advanced)]
        internal static global::System.Globalization.CultureInfo Culture {
            get {
                return resourceCulture;
            }
            set {
                resourceCulture = value;
            }
        }

        /// <summary>
```

```
        ///   Looks up a localized string similar to String1.
        /// </summary>
        internal static string String1 {
            get {
                return ResourceManager.GetString("String1", resourceCulture);
            }
        }
    }
}
```

نکته: فضای نام پیش‌فرض برای منابع کلی در این کلاس‌ها همیشه Resources است که برابر پیشوند (Prefix) عبارت بومی سازی صریح است.

نکته: در کلاس بالا نحوه نمونه سازی کلاس ResourceManager نشان داده شده است. همانطور که مشاهده می‌کنید تعیین کردن مشخصات فایل اصلی Resource مربوطه که در اسمبلی نهایی تولید و کش می‌شود، اجباری است! در مطلب بعدی با این کلاس بیشتر آشنا خواهیم شد.

نکته: همانطور که قبلا نیز اشاره شد، کار تولید اسمبلی مربوط به فایل‌های منابع کلی و محلی و کش کردن آن‌ها در اسمبلی در زمان اجرا کاملا بر عهده ASP.NET است. مثلا در نمونه کد بالا می‌بینید که کلاس ResourceManager برای استخراج نوع Resources.Resource1 از اسمبلی App_GlobalResources نمونه‌سازی شده است، با اینکه این اسمبلی و نوع مذبور در زمان کامپایل و پابلیش وجود ندارد!

برای استفاده از این کلاس می‌توان به صورت زیر عمل کرد:

```
TextBox1.Text = Resources.Resource1.String1;
```

نکته: همانطور که قبلا هم اشاره شد، متاسفانه روش بالا (برخلاف دو متدی که در قسمت قبل توضیح داده شد) به صورت مستقیم از کلاس ResourceManager استفاده می‌کند، که برای بحث سفارشی سازی پرووایدرهای منابع مشکل‌زاست. در مطالب بعدی با معایب آن و نیز راه حل‌های موجود آشنا خواهیم شد.

نکات نهایی

حال که با مفاهیم کلی بیشتری آشنا شدیم بهتر است کمی هم به نکات ریزتر بپردازیم:

نکته: فایل تولیدی توسط ویژوال استودیو در فرایند مدیریت منابع ASP.NET تاثیرگذار **نیست** ! باز هم تاکید می‌کنم که کار استخراج کلیدهای Resource از درون فایلهای resx. کاملا به صورت جداگانه و خودکار و در زمان اجرا انجام می‌شود (درباره این فرایند در مطالب بعدی شرح مفصلی خواهد آمد). درواقع شما می‌توانید خاصیت Custom Tool مربوط به منابع کلی را نیز همانند منابع محلی به رشته‌ای خالی مقداردهی کنید و ببینید که خللی در فرایند مربوطه رخ نخواهد داد!

نکته: تنها برای حالتی که بخواهید از روش آخری که در بالا اشاره شد برای دسترسی با برنامه‌نویسی به منابع کلی بهره ببرید (روش مستقیم)، به این کلاس تولیدی توسط ویژوال استودیو نیاز خواهید داشت. دقت کنید که در این کلاس نیز کار اصلی برعهده کلاس ResourceManager است. درواقع می‌توان کلا از این فایل خودکارتولیدشده صرفنظر کرد و کار استخراج کلیدهای منابع را به صورت مستقیم به نمونه‌ای از کلاس ResourceManager سپرد. این روش نیز در قسمت‌های بعدی شرح داده خواهد شد.

نکته: اگر فایل‌های Resource درون اسمبلی‌های جداگانه‌ای باشند (مثلا در یک پروژه جداگانه، همانطور که در قسمت اول این سری مطالب پیشنهاد شده است)، موتور پیش فرض منابع در ASP.NET بدرد نخواهد خورد! بنابراین یا باید از نمونه‌های اختصاصی کلاس ResourceManager استفاده کرد (کاری که کلاس‌های خودکار تولیدشده توسط ابزارهای ویژوال استودیو انجام می‌دهند)، یا باید از پرووایدرهای سفارشی استفاده کرد که در مطالب بعدی نحوه تولید آن‌ها شرح داده خواهد شد.

همانطور که در ابتدای این مطلب اشاره شد، این مقدمه در اینجا صرفا برای آشنایی بیشتر با این دونوع Resource آورده شده تا ادامه مطلب روشن‌تر باشد، زیرا با توجه به مطالب ارائه شده در قسمت اول این سری، در پروژه‌های MVC استفاده از یک پروژه جداگانه برای نگهداری این منابع راه حل مناسبتری است.

در مطلب بعدی به شرح نحوه تولید پرووایدرهای سفارشی می‌پردازم.

منابع:

http://msdn.microsoft.com/en-us/library/aa905797.aspx
http://msdn.microsoft.com/en-us/library/ms227427.aspx http://www.west-wind.com/presentations/wwdbresourceprovider
http://msdn.microsoft.com/en-us/library/1ztca10y(v=vs.100).aspx
http://msdn.microsoft.com/en-us/library/ms227982(v=vs.100).aspx
http://msdn.microsoft.com/en-us/library/sb6a8618(v=vs.100).aspx

نظرات خوانندگان

نویسنده: میهمان
تاریخ: ۱۳۹۲/۰۲/۱۸ ۱:۳

ممنون از مطلب بسیار مفیدتان

عنوان: **Globalization در ASP.NET MVC - قسمت چهارم**
نویسنده: یوسف نژاد
تاریخ: ۱۳۹۲/۰۲/۲۴ ۲۱:۲۵
آدرس: www.dotnettips.info
برچسب‌ها: Globalization, Resource, .resx, ResourceProviderFactory, ResourceManager

در قسمت قبل مقدمه ای راجع به انواع منابع موجود در ASP.NET و برخی مسائل پیرامون آن ارائه شد. در این قسمت راجع به نحوه رفتار ASP.NET در برخورد با انواع منابع بحث می‌شود.

مدیریت منابع در ASP.NET

در مدل پرووایدر منابع در ASP.NET کار مدیریت منابع از کلاس **ResourceProviderFactory** شروع می‌شود. این کلاس که از نوع abstract تعریف شده است، دو متد برای فراهم کردن پرووایدرهای کلی و محلی دارد.
کلاس پیش‌فرض در ASP.NET برای پیاده‌سازی ResourceProviderFactory در اسمبلی System.Web قرار دارد. این کلاس که **ResXResourceProviderFactory** نام دارد نمونه‌هایی از کلاس‌های **LocalResxResourceProvider** و **GlobalResxResourceProvider** را برمی‌گرداند. درباره این کلاس‌ها در ادامه بیشتر بحث خواهد شد.

نکته: هر سه کلاس پیش‌فرض اشاره شده در بالا و نیز سایر کلاس‌های مربوط به عملیات مدیریت منابع در آن‌ها، همگی در فضای نام System.Web.Compilation قرار دارند و متاسفانه دارای سطح دسترسی internal هستند. بنابراین به صورت مستقیم در دسترس نیستند.

برای نمونه با توجه به تصویر فرضی نشان داده شده در قسمت قبل ، در اولین بارگذاری صفحه SubDir1\Page1.aspx عبارات ضمنی بکاربرده شده در این صفحه برای منابع محلی (در قسمت قبل شرح داده شده است) باعث فراخوانی متد مربوط به Local Resources در کلاس **ResXResourceProviderFactory** می‌شود. این متد نمونه‌ای از کلاس **LocalResXResourceProvider** برمی‌گرداند. (در ادامه با نحوه سفارشی‌سازی این کلاس‌ها نیز آشنا خواهیم شد).
رفتار پیش‌فرض این پرووایدر این است که نمونه‌ای از کلاس **ResourceManager** با توجه به کلید درخواستی برای صفحه موردنظر (مثلا نوع Page1.aspx در اسمبلی App_LocalResources.subdir1.XXXXXX که در تصویر موجود در قسمت قبل نشان داده شده است) تولید می‌کند. حال این کلاس با استفاده از کالچر مربوط به درخواست موردنظر، ورودی موردنظر را از منبع مربوطه استخراج می‌کند. مثلا اگر کالچر موردبحث es (اسپانیایی) باشد، اسمبلی ستلایت موجود در مسیر نسبی \es\ انتخاب می‌شود.
برای روشن‌تر شدن بحث به تصویر زیر که عملیات مدیریت منابع پیش فرض در ASP.NET در درخواست صفحه Page1.aspx از پوشه SubDir1 را نشان می‌دهد، دقت کنید:

همانطور که در قسمت اول این سری مطالب عنوان شد، رفتار کلاس ResourceManager برای یافتن کلیدهای Resource، استخراج آن از نزدیکترین گزینه موجود است. یعنی مثلا برای یافتن کلیدی در کالچر es در مثال بالا، ابتدا اسمبلی‌های مربوط به این کالچر جستجو می‌شود و اگر ورودی موردنظر یافته نشد، جستجو در اسمبلی‌های ستلایت پیش‌فرض سیستم موجود در ریشه فولدر bin برنامه ادامه می‌یابد، تا درنهایت نزدیک‌ترین گزینه پیدا شود (فرایند fallback).

نکته: همانطور که در تصویر بالا نیز مشخص است، نحوه نامگذاری اسمبلی منابع محلی به صورت App_LocalResources.<SubDirectory>.<A random code> است.

نکته: پس از اولین بارگذاری هر اسمبلی، آن اسمبلی به همراه خود نمونه کلاس ResourceManager که مثلا توسط کلاس LocalResXResourceProvider تولید شده است در حافظه سرور کش می‌شوند تا در استفاده‌های بعدی به کار روند.

نکته: فرایند مشابه‌ای برای یافتن کلیدها در منابع کلی (Global Resources) به انجام می‌رسد. تنها تفاوت آن این است که کلاس ResXResourceProviderFactory نمونه‌ای از کلاس **GlobalResXResourceProvider** تولید می‌کند.

چرا پرووایدر سفارشی؟

تا اینجا بالا با کلیات عملیاتی که ASP.NET برای بارگذاری منابع محلی و کلی به انجام می‌رساند، آشنا شدیم. حالا باید به این پرسش پاسخ داد که چرا پرووایدری سفارشی نیاز است؟ علاوه بر دلایلی که در قسمت‌های قبلی به آنها اشاره شد، می‌توان دلایل زیر را نیز برشمرد:

- استفاده از منابع و یا اسمبلی‌های ستلایت موجود - اگر بخواهید در برنامه خود از اسمبلی‌هایی مشترک، بین برنامه‌های ویندوزی و وبی استفاده کنید، و یا بخواهید به هردلیلی از اسمبلی‌های جداگانه‌ای برای این منابع استفاده کنید، مدل پیش‌فرض موجود در ASP.NET جوابگو نخواهد بود.

- استفاده از منابع دیگری به غیر از فایلهای resx. مثل دیتابیس - برای برنامه‌های تحت وب که صفحات بسیار زیاد به همراه ورودی‌های بیشماری از Resourceها دارند، استفاده از مدل پرووایدر منابع پیش‌فرض در ASP.NET و ذخیره تمامی این ورودی‌ها درون فایل‌های resx. بار نسبتا زیادی روی حافظه سرور خواهد گذاشت. درصورت مدیریت بهینه فراخوانی‌های سمت دیتابیس می‌توان با بهره‌برداری از جداول یک دیتابیس به عنوان منبع، کمک زیادی به وب سرور کرد! هم‌چنین با استفاده از دیتابیس می‌توان

مدیریت بهتری بر ورودی‌ها داشت و نیز امکان ذخیره‌سازی حجم بیشتری از داده‌ها در اختیار توسعه دهنده قرار خواهد گرفت. البته به غیر از دیتابیس و فایل‌های resx. نیز گزینه‌های دیگری برای ذخیره‌سازی ورودی‌های این منابع وجود دارند. به عنوان مثال می‌توان مدیریت این منابع را کلا به سیستم دیگری سپرد و درخواست ورودی‌های موردنیاز را به یکسری وب‌سرویس سپرد. برای پیاده سازی چنین سیستمی نیاز است تا مدلی سفارشی تهیه و استفاده شود.

- پیاده سازی امکان به روزرسانی منابع در زمان اجرا - درصورتی‌که بخواهیم امکان بروزرسانی ورودی‌ها را در زمان اجرا در استفاده از فایلهای resx. داشته باشیم، یکی از راه‌حل‌ها، سفارشی سازی این پرووایدرهاست.

مدل پرووایدر منابع

همانطور که قبلا هم اشاره شد، وظیفه استخراج داده‌ها از Resourceها به صورت پیش‌فرض، درنهایت بر عهده نمونه‌ای از کلاس ResourceManager است. در واقع این کلاس کل فرایند انتخاب مناسب‌ترین کلید از منابع موجود را با توجه به کالچر رابط کاربری (UI Culture) در ثرد جاری کپسوله می‌کند. درباره این کلاس در ادامه بیشتر بحث خواهد شد.

هم‌چنین بازهم همانطور که قبلا توضیح داده شد، استفاده از ورودی‌های منابع موجود به دو روش انجام می‌شود. استفاده از عبارات بومی‌سازی و نیز با استفاده از برنامه‌نویسی که ازطریق دومتد GetLocalResourceObject و GetGlobalResourceObject انجام می‌شود. درضمن کلیه عبارات بومی‌سازی در زمان رندر صفحات وب درنهایت تبدیل به فراخوانی‌هایی از این دو متد در کلاس TemplateControl خواهند شد.

عملیات پس از فراخوانی این دو متد جایی است که مدل Resource Provider پیش‌فرض ASP.NET وارد کار می‌شود. این فرایند ابتدا با فراخوانی نمونه‌ای از کلاس ResourceProviderFactory آغاز می‌شود که پیاده‌سازی پیش‌فرض آن در کلاس ResXResourceProviderFactory قرار دارد.

این کلاس سپس با توجه به نوع منبع درخواستی (Global یا Local) نمونه‌ای از پرووایدر مربوطه (که باید اینترفیس IResourceProvider را پیاده‌سازی کرده باشند) را تولید می‌کند. پیاده‌سازی پیش‌فرض این پرووایدرها در ASP.NET در کلاس‌های GlobalResXResourceProvider و LocalResXResourceProvider قرار دارد.

این پروایدرها درنهایت باتوجه به محل ورودی درخواستی، نمونه مناسب از کلاس RsourceManager را تولید و استفاده می‌کنند.

هم‌چنین در پروایدرهای محلی، برای استفاده از عبارات بومی‌سازی ضمنی، نمونه‌ای از کلاس **ResourceReader** مورد استفاده قرار می‌گیرد. در زمان تجزیه و تحلیل صفحه وب درخواستی در سرور، با استفاده از این کلاس کلیدهای موردنظر یافته می‌شوند. این کلاس درواقع پیاده‌سازی اینترفیس IResourceReader بوده که حاوی یک Enumerator که جفت داده‌های Key-Value از کلیدهای Resource را برمی‌گرداند، است.

تصویر زیر نمایی کلی از فرایند پیش‌فرض موردبحث را نشان می‌دهد:

این فرایند باتوجه به پیاده سازی نسبتا جامع آن، قابلیت بسیاری برای توسعه و سفارشی سازی دارد. بنابراین قبل از ادامه مبحث بهتر است، کلاس‌های اصلی این مدل بیشتر شرح داده شوند.

پیاده‌سازی‌ها

کلاس ResourceProviderFactory به صورت زیر تعریف شده است:

```
public abstract class ResourceProviderFactory
{
    public abstract IResourceProvider CreateGlobalResourceProvider(string classKey);
    public abstract IResourceProvider CreateLocalResourceProvider(string virtualPath);
}
```

همانطور که مشاهده می‌کنید دو متد برای تولید پرووایدرهای مخصوص منابع کلی و محلی در این کلاس وجود دارد. پرووایدر کلی تنها نیاز به نام کلید Resource برای یافتن داده موردنظر دارد. اما پرووایدر محلی به مسیر صفحه درخواستی برای اینکار نیاز دارد که با توجه به توضیحات ابتدای این مطلب کاملا بدیهی است.

پس از تولید پرووایدر موردنظر با استفاده از متد مناسب با توجه به شرایط شرح داده شده در بالا، نمونه تولیدشده از کلاس پرووایدر موردنظر وظیفه فراهم‌کردن کلیدهای Resource را برعهده دارد. پرووایدرهای موردبحث باید اینترفیس IResourceProvider را که به صورت زیر تعریف شده است، پیاده سازی کنند:

```
public interface IResourceProvider
{
    IResourceReader ResourceReader { get; }
    object GetObject(string resourceKey, CultureInfo culture);
}
```

همانطور که می‌بینید این پرووایدرها باید یک RsourceReader برای خواندن کلیدهای Resource فراهم کنند. همچنین یک متد با عنوان GetObject که کار اصلی برگرداندن داده ذخیره‌شده در ورودی موردنظر را برعهده دارد باید در این پرووایدرها پیاده‌سازی

شود. همانطور که قبلا اشاره شد، پیاده‌سازی پیش‌فرض این کلاس‌ها درنهایت نمونه‌ای از کلاس ResourceManager را برای یافتن مناسب‌ترین گزینه از بین کلیدهای موجود تولید می‌کند. این نمونه مورد بحث در متد GetObject مورد استفاده قرار می‌گیرد.

نکته: کدهای نشان‌داده‌شده در ادامه مطلب با استفاده از ابزار محبوب ReSharper استخراج شده‌اند. این ابزار برای دریافت این کدها معمولا از APIهای سایت SymbolSource.org استفاده می‌کند. البته منبع اصلی تمام کدهای دات نت فریمورک همان referencesource.microsoft.com است.

کلاس ResXResourceProviderFactory

پیاده‌سازی پیش‌فرض کلاس ResourceProviderFactory در ASP.NET که در کلاس ResXResourceProviderFactory قرار دارد، به صورت زیر است:

```
// Type: System.Web.Compilation.ResXResourceProviderFactory
// Assembly: System.Web, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a
// Assembly location:
C:\Windows\Microsoft.NET\assembly\GAC_32\System.Web\v4.0_4.0.0.0__b03f5f7f11d50a3a\System.Web.dll

using System.Runtime;
using System.Web;
namespace System.Web.Compilation
{
  internal class ResXResourceProviderFactory : ResourceProviderFactory
  {
    [TargetedPatchingOptOut("Performance critical to inline this type of method across NGen image
boundaries")]
    public ResXResourceProviderFactory()    {    }
    public override IResourceProvider CreateGlobalResourceProvider(string classKey)
    {
      return (IResourceProvider) new GlobalResXResourceProvider(classKey);
    }
    public override IResourceProvider CreateLocalResourceProvider(string virtualPath)
    {
      return (IResourceProvider) new LocalResXResourceProvider(VirtualPath.Create(virtualPath));
    }
  }
}
```

در این کلاس برای تولید پرووایدر منابع محلی از کلاس VirtualPath استفاده شده است که امکاناتی جهت استخراج مسیرهای موردنظر با توجه به مسیر نسبی و مجازی ارائه‌شده فراهم می‌کند. متاسفانه این کلاس نیز با سطح دسترسی internal تعریف شده است و امکان استفاده مستقیم از آن وجود ندارد.

کلاس GlobalResXResourceProvider

پیاده‌سازی پیش‌فرض اینترفیس IResourceProvider در ASP.NET برای منابع کلی که در کلاس GlobalResXResourceProvider قرار دارد، به صورت زیر است:

```
internal class GlobalResXResourceProvider : BaseResXResourceProvider
{
  private string _classKey;
  internal GlobalResXResourceProvider(string classKey)
  {
    _classKey = classKey;
  }
  protected override ResourceManager CreateResourceManager()
  {
    string fullClassName = BaseResourcesBuildProvider.DefaultResourcesNamespace + "." + _classKey;
    // If there is no app resource assembly, return null
    if (BuildManager.AppResourcesAssembly == null)
      return null;
    ResourceManager resourceManager = new ResourceManager(fullClassName,
BuildManager.AppResourcesAssembly);
    resourceManager.IgnoreCase = true;
    return resourceManager;
  }
  public override IResourceReader ResourceReader
  {
    get
    {
```

```
      // App resources don't support implicit resources, so the IResourceReader should never be needed
      throw new NotSupportedException();
    }
  }
}
```

در این کلاس عملیات تولید نمونه مناسب از کلاس ResourceManager انجام می‌شود. مقدار BaseResourcesBuildProvider.DefaultResourcesNamespace به صورت زیر تعریف شده است:

```
internal const string DefaultResourcesNamespace = "Resources";
```

که قبلا هم درباره این مقدار پیش فرض اشاره‌ای شده بود.
پارامتر classKey درواقع اشاره به نام فایل اصلی منبع کلی دارد. مثلا اگر این مقدار برابر Resource1 باشد، کلاس ResourceManager برای نوع داده Resources.Resource1 تولید خواهد شد.
هم‌چنین اسمبلی موردنظر برای یافتن ورودی‌های منابع کلی که از BuildManager.AppResourcesAssembly دریافت شده است، به صورت پیش فرض هم‌نام با مسیر منابع کلی و با عنوان App_GlobalResources تولید می‌شود.
کلاس **BuildManager** فرایندهای کامپایل کدها و صفحات برای تولید اسمبلی‌ها و نگهداری از آن‌ها در حافظه را مدیریت می‌کند. این کلاس که محتوای نسبتا مفصلی دارد (نزدیک به 2000 خط کد) به صورت public و sealed تعریف شده است. بنابراین با ریفرنس دادن اسمبلی System.Web در فضای نام System.Web.Compilation در دسترس است، اما نمی‌توان کلاسی از آن مشتق کرد. BuildManager حاوی تعداد زیادی اعضای استاتیک برای دسترسی به اطلاعات اسمبلی‌هاست. اما متاسفانه بیشتر آن‌ها سطح دسترسی عمومی ندارند.

نکته: همانطور که در بالا نیز اشاره شد، ازآنجاکه کلاس ResourceReader در اینجا تنها برای عبارات بومی سازی ضمنی کاربرد دارد، و نیز عبارات بومی‌سازی ضمنی تنها برای منابع محلی کاربرد دارند، در این کلاس برای خاصیت مربوطه در پیاده سازی اینترفیس IResourceProvider یک خطای عدم پشتیبانی (NotSupportedException) صادر شده است.

کلاس LocalResXResourceProvider

پیاده‌سازی پیش‌فرض اینترفیس IResourceProvider در ASP.NET برای منابع محلی که در کلاس LocalResXResourceProvider قرار دارد، به صورت زیر است:

```
internal class LocalResXResourceProvider : BaseResXResourceProvider
{
  private VirtualPath _virtualPath;
  internal LocalResXResourceProvider(VirtualPath virtualPath)
  {
    _virtualPath = virtualPath;
  }
  protected override ResourceManager CreateResourceManager()
  {
    ResourceManager resourceManager = null;
    Assembly pageResAssembly = GetLocalResourceAssembly();
    if (pageResAssembly != null)
    {
      string fileName = _virtualPath.FileName;
      resourceManager = new ResourceManager(fileName, pageResAssembly);
      resourceManager.IgnoreCase = true;
    }
    else
    {
      throw new
InvalidOperationException(SR.GetString(SR.ResourceExpresionBuilder_PageResourceNotFound));
    }
    return resourceManager;
  }
  public override IResourceReader ResourceReader
  {
    get
    {
      // Get the local resource assembly for this page
      Assembly pageResAssembly = GetLocalResourceAssembly();
```

```
      if (pageResAssembly == null) return null;
      // Get the name of the embedded .resource file for this page
      string resourceFileName = _virtualPath.FileName + ".resources";
      // Make it lower case, since GetManifestResourceStream is case sensitive
      resourceFileName = resourceFileName.ToLower(CultureInfo.InvariantCulture);
      // Get the resource stream from the resource assembly
      Stream resourceStream = pageResAssembly.GetManifestResourceStream(resourceFileName);
      // If this page has no resources, return null
      if (resourceStream == null) return null;
      return new ResourceReader(resourceStream);
    }
  }
  [PermissionSet(SecurityAction.Assert, Unrestricted = true)]
  private Assembly GetLocalResourceAssembly()
  {
    // Remove the page file name to get its directory
    VirtualPath virtualDir = _virtualPath.Parent;
    // Get the name of the local resource assembly
    string cacheKey = BuildManager.GetLocalResourcesAssemblyName(virtualDir);
    BuildResult result = BuildManager.GetBuildResultFromCache(cacheKey);
    if (result != null)
    {
      return ((BuildResultCompiledAssembly)result).ResultAssembly;
    }
    return null;
  }
}
```

عملیات موجود در این کلاس باتوجه به فرایندهای مربوط به یافتن اسمبلی مربوطه با استفاده از مسیر ارائه‌شده، کمی پیچیده‌تر از کلاس قبلی است.

در متد GetLocalResourceAssembly عملیات یافتن اسمبلی متناظر با درخواست جاری انجام می‌شود. اینکار باتوجه به نحوه نامگذاری اسمبلی منابع محلی که در ابتدای این مطلب اشاره شد انجام می‌شود. مثلا اگر صفحه درخواستی در مسیر ~/SubDir1/Page1.aspx باشد، در این متد با استفاده از ابزارهای موجود عنوان اسمبلی نهایی برای این مسیر که به صورت App_LocalResources.SubDir1.XXXXX است تولید و درنهایت اسمبلی مربوطه استخراج می‌شود.

درضمن در اینجا هم کلاس ResourceManager برای نوع داده متناظر با نام فایل اصلی منبع محلی تولید می‌شود. مثلا برای مسیر مجازی ~/SubDir1/Page1.aspx نوع داده‌ای با نام Page1.aspx درنظر گرفته خواهد شد (با توجه به نام فایل منبع محلی که باید به صورت Page1.aspx.resx باشد. در قسمت قبل در این باره شرح داده شده است).

نکته: کلاس **SR** (مخفف String Resources) که در فضای نام System.Web قرار دارد، حاوی عناوین کلیدهای Resourceهای مورداستفاده در اسمبلی System.Web است. این کلاس با سطح دسترسی internal و به صورت sealed تعریف شده است. عنوان تمامی کلیدها به صورت ثوابتی از نوع رشته تعریف شده‌اند.

SR درواقع یک Wrapper بر روی کلاس ResourceManager است تا از تکرار عناوین کلیدهای منابع که از نوع رشته هستند، در جاهای مختلف برنامه جلوگیری شود. کار این کلاس مشابه کاری است که کتابخانه T4MVC برای نگهداری عناوین کنترلرها و اکشن‌ها به صورت رشته‌های ثابت انجام می‌دهد. از این روش در جای جای دات نت فریمورک برای نگهداری رشته‌های ثابت استفاده شده است!

نکته: باتوجه به استفاده از عبارات بومی‌سازی ضمنی در استفاده از ورودی‌های منابع محلی، خاصیت ResourceReader در این کلاس نمونه‌ای متناظر برای درخواست جاری از کلاس ResourceReader با استفاده از Stream استخراج شده از اسمبلی یافته شده، تولید می‌کند.

کلاس پایه BaseResXResourceProvider

کلاس پایه BaseResXResourceProvider که در دو پیاده‌سازی نشان داده شده در بالا استفاده شده است (هر دو کلاس از این کلاس مشتق شده‌اند)، به صورت زیر است:

```
internal abstract class BaseResXResourceProvider : IResourceProvider
{
  private ResourceManager _resourceManager;
  ///// IResourceProvider implementation
  public virtual object GetObject(string resourceKey, CultureInfo culture)
  {
```

```
    // Attempt to get the resource manager
    EnsureResourceManager();
    // If we couldn't get a resource manager, return null
    if (_resourceManager == null) return null;
    if (culture == null) culture = CultureInfo.CurrentUICulture;
    return _resourceManager.GetObject(resourceKey, culture);
  }
  public virtual IResourceReader ResourceReader { get { return null; } }
  ///// End of IResourceProvider implementation
  protected abstract ResourceManager CreateResourceManager();
  private void EnsureResourceManager()
  {
    if (_resourceManager != null)  return;
    _resourceManager = CreateResourceManager();
  }
}
```

در این کلاس پیاده‌سازی اصلی اینترفیس IResourceProvider انجام شده است. همانطور که می‌بینید کار نهایی استخراج ورودی‌های منابع در متد GetObject با استفاده از نمونه فراهم شده از کلاس ResourceManager انجام می‌شود.

نکته: دقت کنید که در کد بالا درصورت فراهم نکردن مقداری برای کالچر، از کالچر UI در ثرد جاری (CultureInfo.CurrentUICulture) به عنوان مقدار پیش‌فرض استفاده می‌شود.

کلاس ResourceManager

در زمان اجرا ASP.NET کلید مربوط به منبع موردنظر را با استفاده از کالچر جاری UI انتخاب می‌کند. در قسمت اول این سری مطالب شرح کوتاهی بابت انواع کالچرها داده شد، اما برای توضیحات کاملتر به اینجا مراجعه کنید.
در ASP.NET به صورت پیش‌فرض تمام منابع در زمان اجرا از طریق نمونه‌ای از کلاس ResourceManager در دسترس خواهند بود. به ازای هر نوع Resource که درخواستی برای یک کلید آن ارسال می‌شود یک نمونه از کلاس ResourceManager ساخته می‌شود. در این هنگام (یعنی پس از اولین درخواست به کلیدهای یک منبع) اسمبلی ستلایت مناسب آن پس از یافته شدن (یا تولیدشدن در زمان اجرا) به دامین ASP.NET جاری بارگذاری می‌شود و تا زمانیکه این دامین Unload نشود در حافظه سرور باقی خواهد ماند.

نکته: کلاس ResourceManager **تنها** توانایی استخراج کلیدهای Resource از اسمبلی‌های ستلایتی (فایل‌های resources. که در قسمت اول به آن‌ها اشاره شد) که در AppDomain جاری بارگذاری شده‌اند را دارد.

کلاس ResourceManager به صورت زیر نمونه سازی می‌شود:

```
System.Resources.ResourceManager(string baseName, Assembly assemblyName)
```

پارامتر baseName به نام کامل ریشه اسمبلی اصلی موردنظر(با فضای نام و ...) اما بدون پسوند اسمبلی مربوطه (.resources) اشاره دارد. این نام که برابر نام کلاس نهایی تولیدشده برای منبع موردنظر است همنام با فایل اصلی و پیش‌فرض منبع (فایلی که حاوی عنوان هیچ زبان و کالچری نیست) تولید می‌شود. مثلا برای اسمبلی ستلایت با عنوان MyApplication.MyResource.fa-IR.resources باید از عبارت MyApplication.MyResource استفاده شود.
پارامتر assemblyName نیز به اسمبلی حاوی اسمبلی ستلایت اصلی اشاره دارد. درواقع همان اسمبلی اصلی که نوع داده مربوط به فایل منبع اصلی درون آن embed شده است.
مثلا:

```
var manager = new System.Resources.ResourceManager("Resources.Resource1", typeof(Resource1).Assembly)
```

یا

```
var manager = new System.Resources.ResourceManager("Resources.Resource1",
Assembly.LoadFile(@"c:\MyResources\MyGlobalResources.dll"))
```

روش دیگری نیز برای تولید نمونه‌ای از این کلاس وجود دارد که با استفاده از متد استاتیک زیر که در خود کلاس ResourceManager تعریف شده است انجام می‌شود:

```
public static ResourceManager CreateFileBasedResourceManager(string baseName, string resourceDir, Type
usingResourceSet)
```

در این متد کار استخراج ورودی‌های منابع مستقیما از فایل‌های resources. انجام می‌شود. در اینجا baseName نام فایل اصلی منبع بدون پیشوند resources. است. resourceDir نیز مسیری است که فایل‌های resources. در آن قرار دارند. usingResourceSet نیز نوع کلاس سفارشی سازی شده از ResourceSet برای استفاده به جای کلاس پیش‌فرض است که معمولا مقدار null برای آن وارد می‌شود تا از همان کلاس پیش‌فرض استفاده شود (چون برای بیشتر نیازها همین کلاس پیش‌فرض کفایت می‌کند).
نکته: برای تولید فایل resources. از یک فایل resx. میتوان از ابزار resgen همانند زیر استفاده کرد:

```
resgen d:\MyResources\MyResource.fa.resx
```

نکته: عملیاتی که درون کلاس ResourceManager انجام می‌شود پیچیده‌تر از آن است که به نظر می‌آید. این عملیات شامل فرایندهای بسیاری شامل بارگذاری کلیدهای مختلف یافته شده و مدیریت ذخیره موقت آن‌ها در حافظه (کش)، کنترل و مدیریت انواع Resource Setها، و مهمتر از همه مدیریت عملیات Fallback و ... که در نهایت شامل هزاران خط کد است که با یک جستجوی ساده قابل مشاهده و بررسی است (^).

نمونه‌سازی مناسب از ResourceManager

در کدهای نشان داده شده در بالا برای پیاده‌سازی پیش‌فرض در ASP.NET، مهمترین نکته همان تولید نمونه مناسب از کلاس ResourceManager است. پس از آماده شدن این کلاس عملیات استخراج ورودی‌های منابع براحتی و با مدیریت کامل انجام می‌شود. اما ازآنجاکه تقریبا تمامی APIهای موردنیاز با سطح دسترسی internal تعریف شده‌اند، متاسفانه تهیه و تولید این نمونه مناسب خارج از اسمبلی System.Web به صورت مستقیم وجود ندارد.
درهرصورت، برای آشنایی بیشتر با فرایند نشان داده شده، تولید این نمونه مناسب و استفاده مستقیم از آن می‌تواند مفید و نیز جالب باشد. پس از کمی تحقیق و با استفاده از Reflection به کدهای زیر رسیدم:

```
private ResourceManager CreateGlobalResourceManager(string classKey)
{
  var baseName = "Resources." + classKey;
  var buildManagerType = typeof(BuildManager);
  var property = buildManagerType.GetProperty("AppResourcesAssembly", BindingFlags.Static |
BindingFlags.NonPublic | BindingFlags.GetField);
  var appResourcesAssembly = (Assembly)property.GetValue(null, null);
  return new ResourceManager(baseName, appResourcesAssembly) { IgnoreCase = true };
}
```

تنها نکته کد فوق دسترسی به اسمبلی منابع کلی در خاصیت AppResourcesAssembly از کلاس BuildManager با استفاده از BindingFlagهای نشان داده شده است. نحوه استفاده از این متد هم به صورت زیر است:

```
var manager = CreateGlobalResourceManager("Resource1");
Label1.Text = manager.GetString("String1");
```

اما برای منابع محلی کار کمی پیچیده‌تر است. کد مربوط به تولید نمونه مناسب از ResourceManager برای منابع محلی به صورت زیر خواهد بود:

```
private ResourceManager CreateLocalResourceManager(string virtualPath)
{
  var virtualPathType = typeof(BuildManager).Assembly.GetType("System.Web.VirtualPath", true);
  var virtualPathInstance = Activator.CreateInstance(virtualPathType, BindingFlags.NonPublic |
BindingFlags.Instance, null, new object[] { virtualPath }, CultureInfo.InvariantCulture);
  var buildResultCompiledAssemblyType =
```

```
typeof(BuildManager).Assembly.GetType("System.Web.Compilation.BuildResultCompiledAssembly", true);
  var propertyResultAssembly = buildResultCompiledAssemblyType.GetProperty("ResultAssembly",
BindingFlags.NonPublic | BindingFlags.Instance);
  var methodGetLocalResourcesAssemblyName =
typeof(BuildManager).GetMethod("GetLocalResourcesAssemblyName", BindingFlags.NonPublic |
BindingFlags.Static);
  var methodGetBuildResultFromCache = typeof(BuildManager).GetMethod("GetBuildResultFromCache",
BindingFlags.NonPublic | BindingFlags.Static, null, new Type[] { typeof(string) }, null);

  var fileNameProperty = virtualPathType.GetProperty("FileName");
  var virtualPathFileName = (string)fileNameProperty.GetValue(virtualPathInstance, null);

  var parentProperty = virtualPathType.GetProperty("Parent");
  var virtualPathParent = parentProperty.GetValue(virtualPathInstance, null);

  var localResourceAssemblyName = (string)methodGetLocalResourcesAssemblyName.Invoke(null, new object[]
{ virtualPathParent });
  var buildResultFromCache = methodGetBuildResultFromCache.Invoke(null, new object[] {
localResourceAssemblyName });
  Assembly localResourceAssembly = null;
  if (buildResultFromCache != null)
    localResourceAssembly = (Assembly)propertyResultAssembly.GetValue(buildResultFromCache, null);

  if (localResourceAssembly == null)
    throw new InvalidOperationException("Unable to find the matching resource file.");

  return new ResourceManager(virtualPathFileName, localResourceAssembly) { IgnoreCase = true };
}
```

ازجمله نکات مهم این متد تولید یک نمونه از کلاس VirtualPath برای Parse کردن مسیر مجازی واردشده برای صفحه درخواستی است. از این کلاس برای بدست آوردن نام فایل منبع محلی به همراه مسیر فولدر مربوطه جهت استخراج اسمبلی متناظر استفاده میشود.

نکته مهم دیگر این کد دسترسی به متد GetLocalResourcesAssemblyName از کلاس BuildManager است که با استفاده از مسیر فولدر مربوط به صفحه درخواستی نام اسمبلی منبع محلی مربوطه را برمی‌گرداند.

درنهایت با استفاده از متد GetBuildResultFromCache از کلاس BuildManager اسمبلی موردنظر بدست می‌آید. همانطور که از نام این متد برمی‌آید این اسمبلی از کش خوانده می‌شود. البته مدیریت این اسمبلی‌ها کاملا توسط BuildManager و سایر ابزارهای موجود در ASP.NET انجام خواهد شد.

نحوه استفاده از متد فوق نیز به صورت زیر است:

```
var manager = CreateLocalResourceManager("~/Default.aspx");
Label1.Text = manager.GetString("Label1.Text");
```

نکته: ارائه و شرح کدهای پیاده‌سازی‌های پیش‌فرض برای آشنایی با نحوه صحیح سفارشی سازی این کلاس‌ها آورده شده است. پس با دقت بیشتر بر روی این کدها سعی کنید نحوه پیاده‌سازی مناسب را برای سفارشی‌سازی موردنظر خود پیدا کنید.

تا اینجا با مقدمات فرایند تولید پرووایدرهای سفارشی برای استفاده در فرایند بارگذاری ورودی‌های Resourceها آشنا شدیم. در ادامه به بحث تولید پرووایدرهای سفارشی برای استفاده از دیگر انواع منابع (به غیر از فایل‌های resx.) خواهم پرداخت.

منابع: http://msdn.microsoft.com/en-us/library/aa905797.aspx
http://msdn.microsoft.com/en-us/library/ms227427.aspx http://www.west-wind.com/presentations/wwdbresourceprovider
http://www.codeproject.com/Articles/104667/Under-the-Hood-of-BuildManager-and-Resource-Handli
http://www.onpreinit.com/2009/06/updatable-aspnet-resx-resource-provider.html http://msdn.microsoft.com/en-us/library/h6270d0z(v=vs.100).aspx
http://msdn.microsoft.com/en-us/library/system.web.compilation.resourceproviderfactory.aspx

عنوان: Globalization در ASP.NET MVC - قسمت پنجم
نویسنده: یوسف نژاد
تاریخ: ۱۳۹۲/۰۳/۰۴ ۱۳:۵۰
آدرس: www.dotnettips.info
برچسب‌ها: Localization, Globalization, Internationalization, Resource, ResourceProviderFactory, Custom Resource Provider, Database Resource Provider

در قسمت قبل راجع به مدل پیش‌فرض پرووایدر منابع در ASP.NET بحث نسبتا مفصلی شد. در این قسمت تولید یک پرووایدر سفارشی برای استفاده از دیتابیس به جای فایل‌های resx. به عنوان منبع نگهداری داده‌ها بحث می‌شود.
قبلا هم اشاره شده بود که در پروژه‌های بزرگ ذخیره تمام ورودی‌های منابع درون فایل‌های resx. بازدهی مناسبی نخواهد داشت. هم‌چنین به مرور زمان و با افزایش تعداد این فایل‌ها، کار مدیریت آن‌ها بسیار دشوار و طاقت‌فرسا خواهد شد. درضمن به‌دلیل رفتار سیستم کشینگ این منابع در ASP.NET، که محتویات کل یک فایل را بلافاصله پس از اولین درخواست یکی از ورودی‌های آن در حافظه سرور کش می‌کند، در صورت وجود تعداد زیادی فایل منبع و با ورودی‌های بسیار، با گذشت زمان بازدهی کلی سایت به شدت تحت تاثیر قرار خواهد گرفت.
بنابراین استفاده از یک منبع مثل دیتابیس برای چنین شرایطی و نیز کنترل مدیریت دسترسی به ورودی‌های آن به صورت سفارشی، می‌تواند به بازدهی بهتر برنامه کمک زیادی کند. درضمن فرایند به‌روزرسانی مقادیر این ورودی‌ها در صورت استفاده از یک دیتابیس می‌تواند ساده‌تر از حالت استفاده از فایل‌های resx. انجام شود.

تولید یک پرووایدر منابع دیتابیسی - بخش اول

در بخش اول این مطلب با نحوه پیاده‌سازی کلاس‌های اصلی و اولیه موردنیاز آشنا خواهیم شد. مفاهیم پیشرفته‌تر (مثل کش‌کردن ورودی‌ها و عملیات fallback) و نیز ساختار مناسب جدول یا جداول موردنیاز در دیتابیس و نحوه ذخیره ورودی‌ها برای انواع منابع در دیتابیس در مطلب بعدی آورده می‌شود.
با توجه به توضیحاتی که در قسمت قبل داده شد، می‌توان از طرح اولیه‌ای به صورت زیر برای سفارشی‌سازی یک پرووایدر منابع دیتابیسی استفاده کرد:

اگر مطالب قسمت قبل را خوب مطالعه کرده باشید، پیاده سازی اولیه طرح بالا نباید کار سختی باشد. در ادامه یک نمونه از

پیاده‌سازی‌های ممکن نشان داده شده است.
برای آغاز کار ابتدا یک پروژه ClassLibrary جدید مثلا با نام DbResourceProvider ایجاد کنید و ریفرنسی از اسمبلی System.Web به این پروژه اضافه کنید. سپس کلاس‌هایی که در ادامه شرح داده شده‌اند را به آن اضافه کنید.

کلاس DbResourceProviderFactory

همه چیز از یک ResourceProviderFactory شروع می‌شود. نسخه سفارشی نشان داده شده در زیر برای منابع محلی و کلی از کلاس‌های پرووایدر سفارشی استفاده می‌کند که در ادامه آورده شده‌اند.

```
using System.Web.Compilation;
namespace DbResourceProvider
{
  public class DbResourceProviderFactory : ResourceProviderFactory
  {
    #region Overrides of ResourceProviderFactory
    public override IResourceProvider CreateGlobalResourceProvider(string classKey)
    {
      return new GlobalDbResourceProvider(classKey);
    }
    public override IResourceProvider CreateLocalResourceProvider(string virtualPath)
    {
      return new LocalDbResourceProvider(virtualPath);
    }
    #endregion
  }
}
```

درباره اعضای کلاس ResourceProviderFactory در قسمت قبل توضیحاتی داده شد. در نمونه سفارشی بالا دو متد این کلاس برای برگرداندن پرووایدرهای سفارشی منابع محلی و کلی بازنویسی شده‌اند. سعی شده است تا نمونه‌های سفارشی در اینجا رفتاری همانند نمونه‌های پیش‌فرض در ASP.NET داشته باشند، بنابراین برای پرووایدر منابع کلی (GlobalDbResourceProvider) نام منبع درخواستی (className) و برای پرووایدر منابع محلی (LocalDbResourceProvider) مسیر مجازی درخواستی (virtualPath) به عنوان پارامتر کانستراکتور ارسال می‌شود.
نکته: برای استفاده از این کلاس به جای کلاس پیش‌فرض ASP.NET باید یکسری تنظیمات در فایل کانفیگ برنامه مقصد اعمال کرد که در ادامه آورده شده است.

کلاس BaseDbResourceProvider

برای پیاده‌سازی راحت‌تر کلاس‌های موردنظر، بخش‌های مشترک بین دو پرووایدر محلی و کلی در یک کلاس پایه به صورت زیر قرار داده شده است. این طرح دقیقا مشابه نمونه پیش‌فرض ASP.NET است.

```
using System.Globalization;
using System.Resources;
using System.Web.Compilation;
namespace DbResourceProvider
{
  public abstract class BaseDbResourceProvider : IResourceProvider
  {
    private DbResourceManager _resourceManager;
    protected abstract DbResourceManager CreateResourceManager();
    private void EnsureResourceManager()
    {
      if (_resourceManager != null) return;
      _resourceManager = CreateResourceManager();
    }
    #region Implementation of IResourceProvider
    public object GetObject(string resourceKey, CultureInfo culture)
    {
      EnsureResourceManager();
      if (_resourceManager == null) return null;
      if (culture == null) culture = CultureInfo.CurrentUICulture;
      return _resourceManager.GetObject(resourceKey, culture);
    }
    public virtual IResourceReader ResourceReader { get { return null; } }
    #endregion
  }
}
```

کلاس بالا چون یک کلاس صرفا پایه است بنابراین به صورت abstract تعریف شده است. در این کلاس، از نمونه سفارشی DbResourceManager برای بازیابی داده‌ها از دیتابیس استفاده شده است که در ادامه شرح داده شده است.
در اینجا، از متد CreateResourceManager برای تولید نمونه مناسب از کلاس DbResourceManager استفاده می‌شود. این متد به صورت abstract و protected تعریف شده است بنابراین پیاده‌سازی آن باید در کلاس‌های مشتق شده که در ادامه آورده شده‌اند انجام شود.
در متد EnsureResourceManager کار بررسی نال نبودن resouceManager_ انجام می‌شود تا درصورت نال بودن آن، بلافاصله نمونه‌ای تولید شود.

نکته: ازآنجاکه نقطه آغازین فرایند یعنی تولید نمونه‌ای از کلاس DbResourceProviderFactory توسط خود ASP.NET انجام خواهد شد، بنابراین مدیریت تمام نمونه‌های ساخته شده از کلاس‌هایی که در این مطلب شرح داده می‌شوند درنهایت عملا برعهده ASP.NET است. در ASP.NET درطول عمر یک برنامه تنها یک نمونه از کلاس Factory تولید خواهد شد، و متدهای موجود در آن در حالت عادی تنها یکبار به ازای هر منبع درخواستی (کلی یا محلی) فراخوانی می‌شوند. درنتیجه به ازای هر منبع درخواستی (کلی یا محلی) هر یک از کلاس‌های پرووایدر منابع تنها یک‌بار نمونه‌سازی خواهد شد. بنابراین بررسی نال نبودن این متغیر و تولید نمونه‌ای جدید تنها در صورت نال بودن آن، کاری منطقی است. این نمونه بعدا توسط ASP.NET به ازای هر منبع یا صفحه درخواستی کش می‌شود تا در درخواست‌های بعدی تنها از این نسخه کش‌شده استفاده شود.

در متد GetObject نیز کار استخراج ورودی منابع انجام می‌شود. ابتدا با استفاده از متد EnsureResourceManager از وجود نمونه‌ای از کلاس DbResourceManager اطمینان حاصل می‌شود. سپس درصورتی‌که مقدار این کلاس همچنان نال باشد مقدار نال برگشت داده می‌شود. این حالت وقتی پیش می‌آید که نتوان با استفاده از داده‌های موجود نمونه‌ای مناسب از کلاس DbResourceManager تولید کرد.
سپس مقدار کالچر ورودی بررسی می‌شود و درصورتی‌که نال باشد مقدار کالچر UI ثرد جاری که در CultureInfo.CurrentUICulture قرار دارد برای آن درنظر گرفته می‌شود. درنهایت با فراخوانی متد GetObject از DbResourceManager تولیدی برای کلید و کالچر مربوطه کار استخراج ورودی درخواستی پایان می‌پذیرد.
پراپرتی ResourceReader در این کلاس به صورت virtual تعریف شده است تا بتوان پیاده‌سازی مناسب آن را در هر یک از کلاس‌های مشتق‌شده اعمال کرد. فعلا برای این کلاس پایه مقدار نال برگشت داده می‌شود.

کلاس GlobalDbResourceProvider

برای پرووایدر منابع کلی از این کلاس استفاده می‌شود. نحوه پیاده‌سازی آن نیز دقیقا همانند طرح نمونه پیش‌فرض ASP.NET است.

```
using System;
using System.Resources;
namespace DbResourceProvider
{
  public class GlobalDbResourceProvider : BaseDbResourceProvider
  {
    private readonly string _classKey;
    public GlobalDbResourceProvider(string classKey)
    {
      _classKey = classKey;
    }
    #region Implementation of BaseDbResourceProvider
    protected override DbResourceManager CreateResourceManager()
    {
      return new DbResourceManager(_classKey);
    }
    public override IResourceReader ResourceReader
    {
      get { throw new NotSupportedException(); }
    }
    #endregion
  }
}
```

GlobalDbResourceProvider از کلاس پایه‌ای که در بالا شرح داده شد مشتق شده است. بنابراین تنها بخش‌های موردنیاز یعنی متد CreateResourceManager و پراپرتی ResourceReader در این کلاس پیاده‌سازی شده است.

در اینجا نمونه مخصوص کلاس ResourceManager (همان DbResourceManager) با توجه به نام فایل مربوط به منبع کلی تولید می‌شود. نام فایل در اینجا همان چیزی است که در دیتابیس برای نام منبع مربوطه ذخیره می‌شود. ساختار آن بعدا بحث می‌شود.

همان‌طور که می‌بینید برای پراپرتی ResourceReader خطای عدم پشتیبانی صادر می‌شود. دلیل آن در قسمت قبل و نیز به‌صورت کمی دقیق‌تر در ادامه آورده شده است.

کلاس LocalDbResourceProvider

برای منابع محلی نیز از طرحی مشابه نمونه پیش‌فرض ASP.NET که در قسمت قبل نشان داده شد، استفاده شده است.

```
using System.Resources;
namespace DbResourceProvider
{
  public class LocalDbResourceProvider : BaseDbResourceProvider
  {
    private readonly string _virtualPath;
    public LocalDbResourceProvider(string virtualPath)
    {
      _virtualPath = virtualPath;
    }
    #region Implementation of BaseDbResourceProvider
    protected override DbResourceManager CreateResourceManager()
    {
      return new DbResourceManager(_virtualPath);
    }
    public override IResourceReader ResourceReader
    {
      get { return new DbResourceReader(_virtualPath); }
    }
    #endregion
  }
}
```

این کلاس نیز از کلاس پایه‌ای BaseDbResourceProvider مشتق شده و پیاده‌سازی‌های مخصوص منابع محلی برای متد CreateResourceManager و پراپرتی ResourceReader در آن انجام شده است.
در متد CreateResourceManager کار تولید نمونه‌ای از DbResourceManager با استفاده از مسیر مجازی صفحه درخواستی انجام می‌شود. این فرایند شبیه به پیاده‌سازی پیش‌فرض ASP.NET است. در واقع در پیاده‌سازی جاری، نام منابع محلی همنام با مسیر مجازی متناظر آن‌ها در دیتابیس ذخیره می‌شود. درباره ساختار جدول دیتابیس بعدا بحث می‌شود.
در این کلاس کار بازخوانی کلیدهای موجود برای پراپرتی‌های موجود در یک صفحه از طریق نمونه‌ای از کلاس DbResourceReader انجام شده است. شرح این کلاس در ادامه آمده است.

نکته: همانطور که در قسمت قبل هم اشاره کوتاهی شده بود، از خاصیت ResourceReader در پرووایدر منابع برای تعیین تمام پراپرتی‌های موجود در منبع استفاده می‌شود تا کار جستجوی کلیدهای موردنیاز در عبارات بومی‌سازی **ضمنی** برای رندر صفحه وب راحت‌تر انجام شود. بنابراین از این پراپرتی تنها در پرووایدر منابع **محلی** استفاده می‌شود. ازآنجاکه در عبارات بومی‌سازی ضمنی تنها قسمت اول نام کلید ورودی منبع آورده می‌شود، بنابراین قسمت دوم (و یا قسمت‌های بعدی) کلید موردنظر که همان نام پراپرتی کنترل متناظر است از جستجو میان ورودی‌های یافته شده توسط این پراپرتی بدست می‌آید تا ASP.NET بداند که برای رندر صفحه چه پراپرتی‌هایی نیاز به رجوع به پرووایدر منبع محلی مربوطه دارد (برای آشنایی بیشتر با عبارت بومی‌سازی ضمنی رجوع شود به قسمت قبل).

نکته: دقت کنید که پس از اولین درخواست، خروجی حاصل از enumerator این ResourceReader کش می‌شود تا در درخواست‌های بعدی از آن استفاده شود. بنابراین در حالت عادی، به ازای هر صفحه تنها یکبار این پراپرتی فراخوانده می‌شود. درباره این enumerator در ادامه بحث شده است.

کلاس DbResourceManager

کار اصلی مدیریت و بازیابی ورودی‌های منابع از دیتابیس از طریق کلاس DbResourceManager انجام می‌شود. نمونه‌ای بسیار ساده

و اولیه از این کلاس را در زیر مشاهده می‌کنید:

```
using System.Globalization;
using DbResourceProvider.Data;
namespace DbResourceProvider
{
  public class DbResourceManager
  {
    private readonly string _resourceName;
    public DbResourceManager(string resourceName)
    {
      _resourceName = resourceName;
    }
    public object GetObject(string resourceKey, CultureInfo culture)
    {
      var data = new ResourceData();
      return data.GetResource(_resourceName, resourceKey, culture.Name).Value;
    }
  }
}
```

کار استخراج ورودی‌های منابع با استفاده از نام منبع درخواستی در این کلاس مدیریت خواهد شد. این کلاس با استفاده نام منبع درخواستی به عنوان پارامتر کانستراکتور ساخته می‌شود. با استفاده از متد GetObject که نام کلید ورودی موردنظر و کالچر مربوطه را به عنوان پارامتر ورودی دریافت می‌کند فرایند استخراج انجام می‌شود. برای کپسوله‌سازی عملیات از کلاس جداگانه‌ای (ResourceData) برای تبادل با دیتابیس استفاده شده است. شرح بیشتر درباره این کلاس و نیز پیاده سازی کامل‌تر کلاس DbResourceManager به همراه مدیریت کش ورودی‌های منابع و نیز عملیات fallback در مطلب بعدی آورده می‌شود.

کلاس DbResourceReader

این کلاس که درواقع پیاده‌سازی اینترفیس IResourceReader است برای یافتن تمام کلیدهای تعریف شده برای یک منبع به‌کار می‌رود، پیاده‌سازی آن نیز به صورت زیر است:

```
using System.Collections;
using System.Resources;
using System.Security;
using DbResourceProvider.Data;
namespace DbResourceProvider
{
  public class DbResourceReader : IResourceReader
  {
    private readonly string _resourceName;
    private readonly string _culture;
    public DbResourceReader(string resourceName, string culture = "")
    {
      _resourceName = resourceName;
      _culture = culture;
    }
    #region Implementation of IResourceReader
    public void Close() { }
    public IDictionaryEnumerator GetEnumerator()
    {
      return new DbResourceEnumerator(new ResourceData().GetResources(_resourceName, _culture));
    }
    #endregion
    #region Implementation of IEnumerable
    IEnumerator IEnumerable.GetEnumerator()
    {
      return GetEnumerator();
    }
    #endregion
    #region Implementation of IDisposable
    public void Dispose()
    {
      Close();
    }
    #endregion
  }
}
```

این کلاس تنها با استفاده از نام منبع و عنوان کالچر موردنظر کار بازخوانی ورودی‌های موجود را انجام می‌دهد.

تنها نکته مهم در کد بالا متد GetEnumerator است که نمونه‌ای از اینترفیس IDictionaryEnumerator را برمی‌گرداند. در اینجا از کلاس DbResourceEnumerator که برای کار با دیتابیس طراحی شده، استفاده شده است. همانطور که قبلا هم اشاره شده بود، هر یک از اعضای این enumerator از نوع DictionaryEntry هستند که یک struct است. این کلاس در ادامه شرح داده شده است.
متد Close برای بستن و از بین بردن منابعی است که در تهیه enumerator موردبحث نقش داشته‌اند. مثل منابع شبکه‌ای یا فایلی که باید قبل از اتمام کار با این کلاس به صورت کامل بسته شوند. هرچند در نمونه جاری چنین موردی وجود ندارد و بنابراین این متد بلااستفاده است. در کلاس فوق نیز برای دریافت اطلاعات از ResourceData استفاده شده است که بعدا به همراه ساختار مناسب جدول دیتابیس شرح داده می‌شود.
نکته: دقت کنید که در پیاده‌سازی نشان داده شده برای کلاس LocalDbResourceProvider برای یافتن ورودی‌های موجود از مقدار پیش‌فرض (یعنی رشته خالی) برای کالچر استفاده شده است تا از ورودی‌های پیش‌فرض که در حالت عادی باید شامل تمام موارد تعریف شده موجود هستند استفاده شود (قبلا هم شرح داده شد که منبع اصلی و پیش‌فرض یعنی همانی که برای زبان پیش‌فرض برنامه درنظر گرفته می‌شود و بدون نام کالچر مربوطه است، باید شامل حداکثر ورودی‌های تعریف شده باشد. منابع مربوطه به سایر کالچرها می‌توانند همه این ورودی‌های تعریف‌شده در منبع اصلی و یا قسمتی از آن را شامل شوند. عملیات fallback تضمین می‌دهد که درنهایت نزدیک‌ترین گزینه متناظر با درخواست جاری را برگشت دهد).

کلاس DbResourceEnumerator

کلاس دیگری که در اینجا استفاده شده است، DbResourceEnumerator است. این کلاس در واقع پیاده سازی اینترفیس IDictionaryEnumerator است. محتوای این کلاس در زیر آورده شده است:

```
using System.Collections;
using System.Collections.Generic;
using DbResourceProvider.Models;
namespace DbResourceProvider
{
  public sealed class DbResourceEnumerator : IDictionaryEnumerator
  {
    private readonly List<Resource> _resources;
    private int _dataPosition;
    public DbResourceEnumerator(List<Resource> resources)
    {
      _resources = resources;
      Reset();
    }
    public DictionaryEntry Entry
    {
      get
      {
        var resource = _resources[_dataPosition];
        return new DictionaryEntry(resource.Key, resource.Value);
      }
    }
    public object Key { get { return Entry.Key; } }
    public object Value { get { return Entry.Value; } }
    public object Current { get { return Entry; } }
    public bool MoveNext()
    {
      if (_dataPosition >= _resources.Count - 1) return false;
      ++_dataPosition;
      return true;
    }
    public void Reset()
    {
      _dataPosition = -1;
    }
  }
}
```

تفاوت این اینترفیس با اینترفیس IEnumerable در سه عضو اضافی است که برای استفاده در سیستم مدیریت منابع ASP.NET نیاز است. همان‌طور که در کد بالا مشاهده می‌کنید این سه عضو عبارتند از پراپرتی‌های Entry و Key و Value. پراپرتی Entry که ورودی جاری در enumerator را مشخص می‌کند از نوع DictionaryEntry است. پراپرتی‌های Key و Value هم که از نوع object تعریف شده‌اند برای کلید و مقدار ورودی جاری استفاده می‌شوند.
این کلاس لیستی از Resource به عنوان پارامتر کانستراکتور برای تولید enumerator دریافت می‌کند. کلاس Resource مدل تولیدی از ساختار جدول دیتابیس برای ذخیره ورودی‌های منابع است که در مطلب بعدی شرح داده می‌شود. بقیه قسمت‌های کد فوق هم پیاده‌سازی معمولی یک enumerator است.

نکته: به جای تعریف کلاس جداگانه‌ای برای enumerator اینترفیس IResourceProvider می‌توان از enumerator کلاس‌هایی که IDictionary را پیاده‌سازی کرده‌اند نیز استفاده کرد، مانند کلاس Dictionary<object,object> یا ListDictionary.

تنظیمات فایل کانفیگ

برای اجبار کردن ASP.NET به استفاده از Factory موردنظر باید تنظیمات زیر را در فایل web.config اعمال کرد:

```
<system.web>
    ...
    <globalization resourceProviderFactoryType="نام کامل اسمبلی مربوطه, نام پرووایدر فکتوری به همراه فضای نام آن" />
    ...
</system.web>
```

روش نشان داده شده در بالا حالت کلی تعریف و تنظیم یک نوع داده در فایل کانفیگ را نشان می‌دهد. درباره نام کامل اسمبلی در اینجا شرح داده شده است. مثلا برای پیاده‌سازی نشان داده شده در این مطلب خواهیم داشت:

```
<globalization resourceProviderFactoryType="DbResourceProvider.DbResourceProviderFactory, DbResourceProvider" />
```

در مطلب بعدی درباره ساختار مناسب جدول یا جداول دیتابیس برای ذخیره ورودهای منابع و نیز پیاده‌سازی کامل‌تر کلاس‌های مورداستفاده بحث خواهد شد.

منابع: http://msdn.microsoft.com/en-us/library/aa905797.aspx
http://msdn.microsoft.com/en-us/library/ms227427.aspx http://www.west-wind.com/presentations/wwdbresourceprovider
http://www.onpreinit.com/2009/06/updatable-aspnet-resx-resource-provider.html
http://msdn.microsoft.com/en-us/library/system.web.compilation.resourceproviderfactory.aspx
http://www.codeproject.com/Articles/14190/ASP-NET-2-0-Custom-SQL-Server-ResourceProvider
http://www.codeproject.com/Articles/104667/Under-the-Hood-of-BuildManager-and-Resource-Handli

نظرات خوانندگان

نویسنده: ابوالفضل رجب پور
تاریخ: ۱۳۹۲/۰۳/۰۶ ۱۱:۲۲

سلام جناب یوسف نژاد

برای پروژه م می‌خوام از روند شما استفاده کنم. بی صبرانه منتظر قسمت بعدی هستم

تشکر

عنوان: Globalization در ASP.NET MVC - قسمت ششم
نویسنده: یوسف نژاد
تاریخ: ۱۳۹۲/۰۳/۰۷ ۲۳:۴۰
آدرس: www.dotnettips.info
گروه‌ها: Localization, Globalization, Internationalization, ResourceProviderFactory, Custom Resource Provider, Database Resource Provider, Resource

در قسمت قبل ساختار اصلی و پیاده‌سازی ابتدایی یک پرووایدر سفارشی دیتابیسی شرح داده شد. در این قسمت ادامه بحث و مطالب پیشرفته‌تر آورده شده است.

تولید یک پرووایدر منابع دیتابیسی - بخش دوم

در بخش دوم این سری مطلب، ساختار دیتابیس و مباحث پیشرفته پیاده‌سازی کلاس‌های نشان داده‌شده در بخش اول در قسمت قبل شرح داده می‌شود. این مباحث شامل نحوه کش صحیح و بهینه داده‌های دریافتی از دیتابیس، پیاده‌سازی فرایند fallback، و پیاده‌سازی مناسب کلاس DbResourceManager برای مدیریت کل عملیات است.

ساختار دیتابیس

برای پیاده‌سازی منابع دیتابیسی روش‌های مختلفی برای آرایش جداول جهت ذخیره انواع ورودی‌ها می‌توان درنظر گرفت.
مثلا درصورتی‌که حجم و تعداد منابع بسیار باشد و نیز منابع دیتابیسی به اندازه کافی در دسترس باشد، می‌توان به ازای هر منبع یک جدول درنظر گرفت.
یا درصورتیکه منابع داده‌ای محدودتر باشند می‌توان به ازای هر کالچر یک جدول درنظر گرفت و تمام منابع مربوط به یک کالچر را درون یک جدول ذخیره کرد. درهرصورت نحوه انتخاب آرایش جداول منابع کاملا بستگی به شرایط کاری و سلایق برنامه‌نویسی دارد.
برای مطلب جاری به عنوان یک راه‌حل ساده و کارآمد برای پروژه‌های کوچک و متوسط، تمام ورودی‌های منابع درون یک جدول با ساختاری مانند زیر ذخیره می‌شود:

Column Name	Data Type	Allow Nulls
Id	bigint	☐
Name	nvarchar(200)	☐
[Key]	nvarchar(200)	☐
Culture	nvarchar(6)	☐
Value	nvarchar(MAX)	☐

نام این جدول را با درنظر گرفتن شرایط موجود می‌توان Resources گذاشت.

ستون Name برای ذخیره نام منبع درنظر گرفته شده است. این نام برابر نام منابع درخواستی در سیستم مدیریت منابع ASP.NET است که درواقع برابر همان نام فایل منبع اما بدون پسوند resx. است.

ستون Key برای نگهداری کلید ورودی منبع استفاده می‌شود که دقیقا برابر همان مقداری است که درون فایلهای resx. ذخیره می‌شود.

ستون Culture برای ذخیره کالچر ورودی منبع به کار می‌رود. این مقدار می‌تواند برای کالچر پیش‌فرض برنامه برابر رشته خالی باشد.

ستون Value نیز برای نگهداری مقدار ورودی منبع استفاده می‌شود.

برای ستون Id می‌توان از GUID نیز استفاده کرد. در اینجا برای راحتی کار از نوع داده bigint و خاصیت Identity برای تولید خودکار آن در Sql Server استفاده شده است.

نکته: برای امنیت بیشتر می‌توان یک Unique Constraint بر روی سه فیلد Name و Key و Culture اعمال کرد.

برای نمونه به تصویر زیر که ذخیره تعدای ورودی منبع را درون جدول Resources نمایش می‌دهد دقت کنید:

Id	Name	Key	Culture	Value
1	GlobalTexts	Yes		yesssss
2	GlobalTexts	Yes	fa	بله
3	GlobalTexts	Yes	fr	oui
4	GlobalTexts	No		no
5	GlobalTexts	No	fa	خیر
6	GlobalTexts	No	fr	pas
7	Default.aspx	Label1.Text		Hello
10	Default.aspx	Label1.ForeColor		red
11	Default.aspx	Label1.Text	en-US	hello
13	Default.aspx	Label1.ForeColor	en-US	blue
14	Default.aspx	Label1.Text	fa	درود
16	Default.aspx	Label1.ForeColor	fa	red
17	Default.aspx	Label2.Text		GoodBye
18	Default.aspx	Label2.ForeColor		orange
19	Default.aspx	Label2.Text	en-US	goodbye
20	Default.aspx	Label2.ForeColor	en-US	green
21	dir1/page1.aspx	Label1.Text		sssss
22	dir1/page1.aspx	Label2.Text		aaaaa
23	dir1/page1.aspx	Label1.Text	en-US	String 1
24	dir1/page1.aspx	Label2.Text	en-US	String 2
25	dir1/page1.aspx	Label1.Text	fa	رشته 1
26	dir1/page1.aspx	Label2.Text	fa	رشته 2

اصلاح کلاس DbResourceProviderFactory

برای ذخیره منابع محلی، جهت اطمینان از یکسان بودن نام منبع، متد مربوطه در کلاس DbResourceProviderFactory باید به‌صورت زیر تغییر کند:

```
public override IResourceProvider CreateLocalResourceProvider(string virtualPath)
{
  if (!string.IsNullOrEmpty(virtualPath))
  {
    virtualPath = virtualPath.Remove(0, virtualPath.IndexOf('/') + 1); // removes everything from start
to the first '/'
  }
  return new LocalDbResourceProvider(virtualPath);
}
```

با این تغییر مسیرهای درخواستی چون "~/Default.aspx" و یا "/Default.aspx" هر دو به صورت "Default.aspx" در می‌آیند تا با نام ذخیره شده در دیتابیس یکسان شوند.

ارتباط با دیتابیس

خوشبختانه برای تبادل اطلاعات با جدول بالا امروزه راه‌های زیادی وجود دارد. برای پیاده‌سازی آن مثلا می‌توان از یک اینترفیس استفاده کرد. سپس با استفاده از سازوکارهای موجود مثلا به‌کارگیری IoC ، نمونه مناسبی از پیاده‌سازی اینترفیس مذبور را در اختیار برنامه قرار داد.
اما برای جلوگیری از پیچیدگی بیش از حد و دور شدن از مبحث اصلی، برای پیاده‌سازی فعلی از EF Code First به صورت مستقیم در پروژه استفاده شده است که سری آموزشی کاملی از آن در همین سایت وجود دارد.

پس از پیاده‌سازی کلاس‌های مرتبط برای استفاده از EF Code First، از کلاس ResourceData که در بخش اول نیز نشان داده شده بود، برای کپسوله کردن ارتباط با داده‌ها استفاده می‌شود که نمونه‌ای ابتدایی از آن در زیر آورده شده است:

```
using System.Collections.Generic;
using System.Linq;
using DbResourceProvider.Models;

namespace DbResourceProvider.Data
{
  public class ResourceData
  {
    private readonly string _resourceName;
    public ResourceData(string resourceName)
    {
      _resourceName = resourceName;
    }
    public Resource GetResource(string resourceKey, string culture)
    {
      using (var data = new TestContext())
      {
        return data.Resources.SingleOrDefault(r => r.Name == _resourceName && r.Key == resourceKey &&
r.Culture == culture);
      }
    }
    public List<Resource> GetResources(string culture)
    {
      using (var data = new TestContext())
      {
        return data.Resources.Where(r => r.Name == _resourceName && r.Culture == culture).ToList();
      }
    }
  }
}
```

کلاس فوق نسبت به نمونه‌ای که در قسمت قبل نشان داده شد کمی فرق دارد. بدین صورت که برای راحتی بیشتر نام منبع

درخواستی به جای پارامتر متدها، در اینجا به عنوان پارامتر کانستراکتور وارد می‌شود.

نکته: درصورتی‌که این کلاس‌ها در پروژه‌ای جداگانه قرار دارند، باید ConnectionString مربوطه در فایل کانفیگ برنامه مقصد نیز تنظیم شود.

کش کردن ورودی‌ها

برای کش کردن ورودی‌ها این نکته را که قبلا هم به آن اشاره شده بود باید درنظر داشت:

پس از اولین درخواست برای هر منبع، نمونه تولیدشده از پرووایدر مربوطه در حافظه سرور کش خواهد شد.

یعنی متدهای کلاس DbResourceProviderFactory **به‌ازای هر منبع** تنها یکبار فراخوانی می‌شود. نمونه‌های کش‌شده از پرووایدرهای کلی و محلی به همراه تمام محتویاتشان (مثلا نمونه تولیدی از کلاس DbResourceManager) تا زمان Unload شدن سایت در حافظه سرور باقی می‌مانند. بنابراین عملیات کشینگ ورودی‌ها را می‌توان درون خود کلاس DbResourceManager به ازای هر منبع انجام داد.

برای کش کردن ورودی‌های هر منبع می‌توان چند روش را درپیش گرفت. روش اول این است که به ازای هر کلید درخواستی تنها ورودی مربوطه از دیتابیس فراخوانی شده و در برنامه کش شود. این روش برای حالاتی که تعداد ورودی‌ها یا تعداد درخواست‌های کلیدهای هر منبع کم باشد مناسب خواهد بود.

یکی از پیاده‌سازی این روش این است که ورودی‌ها به ازای هر کالچر ذخیره شوند. پیاده‌سازی اولیه این نوع فرایند کشینگ در کلاس DbResourceManager به صورت زیر است:

```
using System.Collections.Generic;
using System.Globalization;
using DbResourceProvider.Data;
namespace DbResourceProvider
{
  public class DbResourceManager
  {
    private readonly string _resourceName;
    private readonly Dictionary<string, Dictionary<string, object>> _resourceCacheByCulture;
    public DbResourceManager(string resourceName)
    {
      _resourceName = resourceName;
      _resourceCacheByCulture = new Dictionary<string, Dictionary<string, object>>();
    }
    public object GetObject(string resourceKey, CultureInfo culture)
    {
      return GetCachedObject(resourceKey, culture.Name);
    }
    private object GetCachedObject(string resourceKey, string cultureName)
    {
      if (!_resourceCacheByCulture.ContainsKey(cultureName))
        _resourceCacheByCulture.Add(cultureName, new Dictionary<string, object>());
      var cachedResource = _resourceCacheByCulture[cultureName];
      lock (this)
      {
        if (!cachedResource.ContainsKey(resourceKey))
        {
          var data = new ResourceData(_resourceName);
          var dbResource = data.GetResource(resourceKey, cultureName);
          if (dbResource == null) return null;
          var cachedResources = _resourceCacheByCulture[cultureName];
          cachedResources.Add(dbResource.Key, dbResource.Value);
        }
      }
      return cachedResource[resourceKey];
    }
  }
}
```

همانطور که قبلا توضیح داده شد کشِ پرووایدرهای منابع به ازای هر منبع درخواستی (و به تبع آن نمونه‌های موجود در آن مثل DbResourceManager) برعهده خود ASP.NET است. بنابراین برای کش کردن ورودی‌های درخواستی هر منبع در کلاس DbResourceManager تنها کافی است آن‌ها را درون یک متغیر محلی در سطح کلاس (فیلد) ذخیره کرد. کاری که در کد بالا در متغیر _resourceCacheByCulture انجام شده است. در این متغیر که از نوع دیکشنری تعریف شده است کلیدهای هر عضو آن برابر نام کالچر مربوطه است. مقادیر هر عضو این دیکشنری نیز خود یک دیکشنری است که ورودی‌های منابع مربوط به کالچر مربوطه در

آن ذخیره می‌شوند.
عملیات در متد GetCachedObject انجام می‌شود. همان‌طور که می‌بینید ابتدا وجود ورودی موردنظر در متغیر کشینگ بررسی می‌شود و درصورت عدم وجود، مقدار آن مستقیما از دیتابیس درخواست می‌شود. سپس این مقدار درخواستی ابتدا درون متغیر کشینگ ذخیره شده (به همراه بلاک lock) و درنهایت برگشت داده می‌شود.

نکته: کل فرایند بررسی وجود کلید در متغیر کشینگ (شرط دوم در متد GetCachedObject) درون بلاک lock قرار داده شده است تا در درخواست‌های همزمان احتمال افزودن چندباره یک کلید ازبین برود.

پیاده‌سازی دیگر این فرایند کشینگ، ذخیره ورودی‌ها براساس نام کلید به جای نام کالچر است. یعنی کلید دیکشنری اصلی نام کلید و کلید دیکشنری داخلی نام کالچر است که این روش زیاد جالب نیست.
روش دوم که بیشتر برای برنامه‌های بزرگ با ورودی‌ها و درخواست‌های زیاد به‌کار می‌رود این است که درهر بار درخواست به دیتابیس به جای دریافت تنها همان ورودی درخواستی، تمام ورودی‌های منبع و کالچر درخواستی استخراج شده و کش می‌شود تا تعداد درخواست‌های به سمت دیتابیس کاهش یابد. برای پیاده‌سازی این روش کافی است تغییرات زیر در متد GetCachedObject اعمال شود:

```
private object GetCachedObject(string resourceKey, string cultureName)
{
  lock (this)
  {
    if (!_resourceCacheByCulture.ContainsKey(cultureName))
    {
      _resourceCacheByCulture.Add(cultureName, new Dictionary<string, object>());
      var cachedResources = _resourceCacheByCulture[cultureName];
      var data = new ResourceData(_resourceName);
      var dbResources = data.GetResources(cultureName);
      foreach (var dbResource in dbResources)
      {
        cachedResources.Add(dbResource.Key, dbResource.Value);
      }
    }
  }
  var cachedResource = _resourceCacheByCulture[cultureName];
  return !cachedResource.ContainsKey(resourceKey) ? null : cachedResource[resourceKey];
}
```

دراینجا هم می‌توان به جای استفاده از نام کالچر برای کلید دیکشنری اصلی از نام کلید ورودی منبع استفاده کرد که چندان توصیه نمی‌شود.

نکته: انتخاب یکی از دو روش فوق برای فرایند کشینگ کاملا به **شرایط موجود** و سلیقه برنامه نویس بستگی دارد.

فرایند Fallback

درباره فرایند fallback به اندازه کافی در قسمت‌های قبلی توضیح داده شده است. برای پیاده‌سازی این فرایند ابتدا باید به نوعی به سلسله مراتب کالچرهای موجود از کالچر جاری تا کالچر اصلی و پیش فرض سیستم دسترسی پیدا کرد. برای اینکار ابتدا باید با استفاده از روشی کالچر والد یک کالچر را بدست آورد. کالچر والد کالچری است که عمومیت بیشتری نسبت به کالچر موردنظر دارد. مثلا کالچر fa، کالچر والد fa-IR است. همچنین کالچر Invariant به عنوان والد تمام کالچرها شناخته می‌شود.
خوشبختانه در کلاس CultureInfo (که در قسمت‌های قبلی شرح داده شده است) یک پراپرتی با عنوان Parent وجود دارد که کالچر والد را برمی‌گرداند.
برای رسیدن به سلسله مراتب مذبور در کلاس ResourceManager دات نت، از کلاسی با عنوان ResourceFallbackManager استفاده می‌شود. هرچند این کلاس با سطح دسترسی internal تعریف شده است اما نام‌گذاری نامناسبی دارد زیرا کاری که می‌کند به عنوان Manager هیچ ربطی ندارد. این کلاس با استفاده از یک کالچر ورودی، یک enumerator از سلسله مراتب کالچرها که در بالا صحبت شد تهیه می‌کند.
با استفاده پیاده‌سازی موجود در کلاس ResourceFallbackManager کلاسی با عنوان CultureFallbackProvider تهیه کردم که به صورت زیر است:

```
using System.Collections;
using System.Collections.Generic;
using System.Globalization;
namespace DbResourceProvider
{
  public class CultureFallbackProvider : IEnumerable<CultureInfo>
  {
    private readonly CultureInfo _startingCulture;
    private readonly CultureInfo _neutralCulture;
    private readonly bool _tryParentCulture;
    public CultureFallbackProvider(CultureInfo startingCulture = null,
                                   CultureInfo neutralCulture = null,
                                   bool tryParentCulture = true)
    {
      _startingCulture = startingCulture ?? CultureInfo.CurrentUICulture;
      _neutralCulture = neutralCulture;
      _tryParentCulture = tryParentCulture;
    }
    #region Implementation of IEnumerable<CultureInfo>
    public IEnumerator<CultureInfo> GetEnumerator()
    {
      var reachedNeutralCulture = false;
      var currentCulture = _startingCulture;
      do
      {
        if (_neutralCulture != null && currentCulture.Name == _neutralCulture.Name)
        {
          yield return CultureInfo.InvariantCulture;
          reachedNeutralCulture = true;
          break;
        }
        yield return currentCulture;
        currentCulture = currentCulture.Parent;
      } while (_tryParentCulture && !HasInvariantCultureName(currentCulture));
      if (!_tryParentCulture || HasInvariantCultureName(_startingCulture) || reachedNeutralCulture)
        yield break;
      yield return CultureInfo.InvariantCulture;
    }
    #endregion
    #region Implementation of IEnumerable
    IEnumerator IEnumerable.GetEnumerator()
    {
      return GetEnumerator();
    }
    #endregion
    private bool HasInvariantCultureName(CultureInfo culture)
    {
      return culture.Name == CultureInfo.InvariantCulture.Name;
    }
  }
}
```

این کلاس که اینترفیس IEnumerable<CultureInfo> را پیاده‌سازی کرده است، سه پارامتر کانستراکتور دارد.

اولین پارامتر، کالچر جاری یا آغازین را مشخص می‌کند. این کالچری است که تولید enumerator مربوطه از آن آغاز می‌شود. درصورتی‌که این پارامتر نال باشد مقدار کالچر UI در ثرد جاری برای آن درنظر گرفته می‌شود. مقدار پیش‌فرضی که برای این پارامتر درنظر گرفته شده است، null است.

پارامتر بعدی کالچر خنثی موردنظر کاربر است. این کالچری است که درصورت رسیدن enumerator به آن کار پایان خواهد یافت. درواقع کالچر پایانی enumerator است. این پارامتر می‌تواند نال باشد. مقدار پیش‌فرضی که برای این پارامتر درنظر گرفته شده است، null است.

پارمتر آخر هم تعیین می‌کند که آیا enumerator از کالچرهای والد استفاده بکند یا خیر. مقدار پیش‌فرضی که برای این پارامتر درنظر گرفته شده است، true است.

کار اصلی کلاس فوق در متد GetEnumerator انجام می‌شود. در این کلاس یک حلقه do-while وجود دارد که enumerator را با استفاده از کلمه کلیدی yield تولید می‌کند. در این متد ابتدا درصورت نال نبودن کالچر خنثی ورودی، بررسی می‌شود که آیا نام کالچر جاری حلقه (که در متغیر محلی currentCulture ذخیره شده است) برابر نام کالچر خنثی است یا خیر. درصورت برقراری شرط، کار این حلقه با برگشت CultureInfo.InvariantCulture پایان می‌یابد. InvariantCulture کالچر بدون زبان و فرهنگ و موقعیت مکانی است که درواقع به عنوان کالچر والد تمام کالچرها درنظر گرفته می‌شود. پراپرتی Name این کالچر برابر string.Empty است.

کار حلقه با برگشت مقدار کالچر جاری enumerator ادامه می‌یابد. سپس کالچر جاری با کالچر والدش مقداردهی می‌شود. شرط قسمت while حلقه تعیین می‌کند که درصورتی‌که کلاس برای استفاده از کالچرهای والد تنظیم شده باشد، تا زمانی که نام کالچر جاری برابر نام کالچر Invariant **نباشد** ، تولید اعضای enumerator ادامه یابد.
درانتها نیز درصورتی‌که با شرایط موجود، قبلا کالچر Invariant برگشت داده نشده باشد این کالچر نیز yield می‌شود. درواقع درصورتی‌که استفاده از کالچرهای والد اجازه داده **نشده** باشد **یا** کالچر آغازین برابر کالچر Invariant باشد و **یا** قبلا به دلیل رسیدن به کالچر خنثی ورودی، مقدار کالچر Invariant برگشت داده شده باشد، enumerator قطع شده و عملیات پایان می‌یابد. در غیر اینصورت کالچر Invariant به عنوان کالچر پایانی برگشت داده می‌شود.

استفاده از CultureFallbackProvider

با استفاده از کلاس CultureFallbackProvider می‌توان عملیات جستجوی ورودی‌های درخواستی را با ترتیبی مناسب بین تمام کالچرهای موجود به انجام رسانید.
برای استفاده از این کلاس باید تغییراتی در متد GetObject کلاس DbResourceManager به صورت زیر اعمال کرد:

```
public object GetObject(string resourceKey, CultureInfo culture)
{
  foreach (var currentCulture in new CultureFallbackProvider(culture))
  {
    var value = GetCachedObject(resourceKey, currentCulture.Name);
    if (value != null) return value;
  }
  throw new KeyNotFoundException("The specified 'resourceKey' not found.");
}
```

با استفاده از یک حلقه foreach درون enumerator کلاس CultureFallbackProvider، کالچرهای موردنیاز برای fallback یافته می‌شوند. در اینجا از مقادیر پیش‌فرض دو پارامتر دیگر کانستراکتور کلاس CultureFallbackProvider استفاده شده است.
سپس به ازای هر کالچر یافته شده مقدار ورودی درخواستی بدست آمده و درصورتی‌که نال نباشد (یعنی ورودی موردنظر برای کالچر جاری یافته شود) آن مقدار برگشت داده می‌شود و درصورتی‌که نال باشد عملیات برای کالچر بعدی ادامه می‌یابد.
درصورتی‌که ورودی درخواستی یافته نشود (خروج از حلقه بدون برگشت مقداری برای ورودی منبع درخواستی) استثنای KeyNotFoundException صادر می‌شود تا کاربر را از اشتباه رخداده مطلع سازد.

آزمایش پرووایدر سفارشی

ابتدا تنظیمات موردنیاز فایل کانفیگ را که در قسمت قبل نشان داده شد، در برنامه خود اعمال کنید.

داده‌های نمونه نشان داده شده در ابتدای این مطلب را درنظر بگیرید. حال اگر در یک برنامه وب اپلیکیشن، صفحه Default.aspx در ریشه سایت حاوی دو کنترل زیر باشد:

```
<asp:Label ID="Label1" runat="server" meta:resourcekey="Label1" />
<asp:Label ID="Label2" runat="server" meta:resourcekey="Label2" />
```

خروجی برای کالچر "en-US" (معمولا پیش‌فرض، اگر تنظیمات سیستم عامل تغییر نکرده باشد) چیزی شبیه تصویر زیر خواهد بود:

hello goodbye

سپس تغییر زیر را در فایل web.config اعمال کنید تا کالچر UI سایت به fa تغییر یابد (به بخش uiCulture="fa" دقت کنید):

```
<globalization uiCulture="fa" resourceProviderFactoryType =
```

```
"DbResourceProvider.DbResourceProviderFactory, DbResourceProvider" />
```

بنابراین صفحه Default.aspx با همان داده‌های نشان داده شده در بالا به صورت زیر تغییر خواهد کرد:

GoodBye درود

می‌بینید که با توجه به عدم وجود مقداری برای Label2.Text برای کالچر fa، عملیات fallback اتفاق افتاده است.

بحث و نتیجه‌گیری

کار تولید یک پرووایدر منابع سفارشی دیتابیسی به اتمام رسید. تا اینجا اصول کلی تولید یک پرووایدر سفارشی شرح داده شد. بدین ترتیب می‌توان برای هر حالت خاص دیگری نیز پرووایدرهای سفارشی مخصوص ساخت تا مدیریت منابع به آسانی تحت کنترل برنامه نویس قرار گیرد.

اما نکته‌ای را که باید به آن توجه کنید این است که در پیاده‌سازی‌های نشان داده شده با توجه به نحوه کش‌شدن مقادیر ورودی‌ها، اگر این مقادیر در دیتابیس تغییر کنند، تا زمانیکه سایت ریست نشود این تغییرات در برنامه اعمال نخواهد شد. زیرا همانطور که اشاره شد، مدیریت نمونه‌های تولیدشده از پرووایدرهای منابع برای هر منبع درخواستی درنهایت برعهده ASP.NET است. بنابراین باید مکانیزمی پیاده شود تا کلاس DbResourceManager از به‌روزرسانی ورودی‌های کش‌شده اطلاع یابد تا آنها را ریفرش کند.

در ادامه درباره روش‌های مختلف نحوه پیاده‌سازی قابلیت به‌روزرسانی ورودی‌های منابع در زمان اجرا با استفاده از پرووایدرهای منابع سفارشی بحث خواهد شد. همچنین راه‌حل‌های مختلف استفاده از این پرووایدرهای سفارشی در جاهای مختلف پروژه‌های MVC شرح داده می‌شود.

البته مباحث پیشرفته‌تری چون تزریق وابستگی برای پیاده‌سازی لایه ارتباط با دیتابیس در بیرون و یا تولید یک Factory برای تزریق کامل پرووایدر منابع از بیرون نیز جای بحث و بررسی دارد.

منابع

http://weblogs.asp.net/thangchung/archive/2010/06/25/extending-resource-provider-for-soring-resources-in-the-database.aspx

http://msdn.microsoft.com/en-us/library/aa905797.aspx

http://msdn.microsoft.com/en-us/library/system.web.compilation.resourceproviderfactory.aspx

http://www.dotnetframework.org/default.aspx/.../ResourceFallbackManager@cs

http://www.codeproject.com/Articles/14190/ASP-NET-2-0-Custom-SQL-Server-ResourceProvider

http://www.west-wind.com/presentations/wwdbresourceprovider

نظرات خوانندگان

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۰۳/۰۸ ۰:۴۲

با تشکر از کار زیبای شما
لطفا برچسب resource را اضافه کنید تا پیوستگی مطالب حفظ شود.

نویسنده: یوسف نژاد
تاریخ: ۱۳۹۲/۰۳/۰۸ ۱:۴۰

با تشکر از دقت نظر شما.
برچسب Resource هم اضافه شد.

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۰۳/۰۸ ۳:۱۵

مهندس یک سوال؟
مشکلی نداره ما سه جدول:
1- جدولی برای ذخیره نام کالچرها
2-جدولی برای ذخیره عنوان کلیدهای اصلی
3- جدولی برای ذخیره مقادیر یک کالچر برای یک کلید خاص

تعریف کنیم؟
اگر درست فهمیده باشم فقط باید بخش بازیابی کلیدها تغییر کنه درسته؟

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۳/۰۸ ۸:۴۱

اون وقت حداقل 2 تا join باید بنویسید و وجود هر join یعنی کم‌تر شدن سرعت دسترسی به اطلاعات. چرا؟ چه تکرار اطلاعاتی رو مشاهده می‌کنید که قصد دارید تا این حد نرمالش کنید؟ نام و کلید و فرهنگ یک موجودیت هستند.

نویسنده: یوسف نژاد
تاریخ: ۱۳۹۲/۰۳/۰۸ ۹:۱۱

دلیل خاصی برای تفکیک این چنینی وجود نداره و همونطور که دوستمون گفتن این روشی که شما اشاره کردین مشکلات و معایبی هم به همراه داره.
روش اشاره شده تو این مطلب تو بیش از 99 درصد پروژه‌ها کفایت میکنه. فقط تو پروژه‌های بسیار بسیار بزرگ با ورودی‌های منابع بسیار بسیار زیاد (چند صد هزار و یا بیشتر) تغییر این ساختار برای رسیدن به کارایی مناسب میتونه مفید باشه.
درهرصورت اگر نیاز به تغییر ساختار جدول دارین فقط لایه دسترسی به بانک باید تغییر بکنه و فرایند کلی دسترسی به ورودی‌های منابع ذخیره شده در دیتابیس باید به همون صورتی باشه که در اینجا آورده شده. یعنی درنهایت با استفاده از سه پارامتر نام منبع، نام کالچر و عنوان کلید درخواستی کار استخراج مقدار ورودی باید انجام بشه.

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۰۳/۰۸ ۱۰:۱۴

برای طراحی یک سامانه مدیریت محتوا با کلی ماژول فکر می‌کنم حرفم منطقی باشه مهندس، در ضمن همونجوری که مهندس یوسف نژاد فرمودن اطلاعات در بازیابی اولیه کش میشه و تا ری ستارت شدن سایت در حافظه می‌مونه، فکر می‌کنم چندان تاثیری

بروی کارایی داشته باشه با توجه به فرضیات، فرض کن من 10000 عنوان دارم، 30 تا زبان دارم در این صورت توی یک جدول زبان انگلیسی (en-کالچر انگلیسی) 10000 بار تکرار میشه علاوه بر اون عنوان مثلا "نام کاربری" به ازای 30 زبان 30 بار تکرار میشه زیادم حرف من غیر منطقی نیست و الا حرف شما درسته بله join سرعت پایین میاره اما ما که قرار نیست زیادی دسترسی به این جداول داشته باشیم.

"پس از اولین درخواست برای هر منبع، نمونه تولیدشده از پرووایدر مربوطه در حافظه سرور کش خواهد شد. " سخن مهندس یوسف نژاد

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۳/۰۸ ۱۲:۱۰

یک سری از برآوردها تخیلی هستند. حتی مایکروسافت هم با لشگر مترجم‌هایی که داره مثلا برای شیرپوینت تجاری خودش زیر 10 تا زبان رو تونسته ارائه بده.

نویسنده: بهنام حقی
تاریخ: ۱۳۹۳/۰۱/۳۱ ۱۷:۹

با سلام
من این حالت رو میخوام با uow میخوام پیاده سازی کنم. میخوام یک سری تغییرات تو ساختار جدول بدم.
یک جدول برای مدیریت اضافه و حذف زبان (نام، RTL، ISO، Culture و ...) و جدول دیگم برای ریسورس ها(کلید، اسم، مقدار)
در واقع میخوام مقادیر ریسورس‌ها با اضافه و حذف شدن یک زبان به سیستم مدیریت بشه.
میخواستم ببینم که چه پیشنهادی برای این حالت دارید؟

عنوان: Globalization در ASP.NET MVC - قسمت هفتم
نویسنده: یوسف نژاد
تاریخ: ۱۳۹۲/۰۳/۱۲ ۱۳:۳۰
آدرس: www.dotnettips.info
گروه‌ها: Localization, Globalization, Internationalization, Resource, Custom Resource Provider, Database Resource Provider

در قسمت قبل مطالب تکمیلی تولید پرووایدر سفارشی منابع دیتابیسی ارائه شد. در این قسمت نحوه بروزرسانی ورودی‌های منابع در زمان اجرا بحث می‌شود.

تولید یک پرووایدر منابع دیتابیسی - بخش سوم

برای پیاده‌سازی ویژگی به‌روزرسانی ورودی‌های منابع در زمان اجرا راه‌حل‌های مختلفی ممکن است به ذهن برنامه‌نویس خطور کند که هر کدام معایب و مزایای خودش را دارد. اما درنهایت بسته به شرایط موجود انتخاب روش مناسب برعهده خود برنامه‌نویس است.

مثلا برای پرووایدر سفارشی دیتابیسی تهیه‌شده در مطالب قبلی، تنها کافی است ابزاری تهیه شود تا به کاربران اجازه به‌روزرسانی مقادیر موردنظرشان در دیتابیس را بدهد که کاری بسیار ساده است. بدین ترتیب به‌روزرسانی این مقادیر در زمان اجرا کاری بسیار ابتدایی به نظر می‌رسد. اما در قسمت قبل نشان داده شد که برای بالا بردن بازدهی بهتر است که مقادیر موجود در دیتابیس در حافظه سرور کش شوند. استراتژی اولیه و ساده‌ای نیز برای نحوه پیاده‌سازی این فرایند کشینگ ارائه شد. بنابراین باید امکاناتی فراهم شود تا درصورت تغییر مقادیر کش‌شده در سمت دیتابیس، برنامه از این تغییرات آگاه شده و نسبت به به‌روزرسانی این مقادیر در متغیر کشینگ اقدامات لازم را انجام دهد.

اما همان‌طور که در قسمت قبل نیز اشاره شد، نکته‌ای که باید درنظر داشت این است که مدیریت تمامی نمونه‌های تولیدشده از کلاس‌های موردبحث کاملا برعهده ASP.NET است، بنابراین دسترسی مستقیمی به این نمونه‌ها در بیرون و در زمان اجرا وجود ندارد تا این ویژگی را بتوان در مورد آن‌ها پیاده کرد.

یکی از روش‌های موجود برای حل این مشکل این است که مکانیزمی پیاده شود تا بتوان به تمامی نمونه‌های تولیدی از کلاس DbResourceManager در بیرون از محیط سیستم مدیریت منابع ASP.NET دسترسی داشت. مثلا یک کلاس حاول متغیری استاتیک جهت ذخیره نمونه‌های تولیدی از کلاس DbResourceManager، به کتابخانه خود اضافه کرد تا با استفاده از یکسری امکانات بتوان این نمونه‌های تولیدی را از تغییرات رخداده در سمت دیتابیس آگاه کرد. در این قسمت پیاده‌سازی این راه‌حل شرح داده می‌شود.

.

نکته: قبل از هرچیز برای مناسب شدن طراحی کتابخانه تولیدی و افزایش امنیت آن بهتر است تا سطح دسترسی تمامی کلاس‌های پیاده‌سازی شده تا این مرحله به internal تغییر کند. ازآنجاکه سیستم مدیریت منابع ASP.NET از ریفلکشن برای تولید نمونه‌های موردنیاز خود استفاده می‌کند، بنابراین این تغییر تاثیری بر روند کاری آن نخواهد گذاشت.

.

نکته: با توجه به شرایط خاص موجود، ممکن است نام‌های استفاده شده برای کلاس‌های این کتابخانه کمی گیج‌کننده باشد. پس با دقت بیشتری به مطلب توجه کنید.

.

پیاده‌سازی امکان پاک‌سازی مقادیر کش‌شده

برای این‌کار باید تغییراتی در کلاس DbResourceManager داده شود تا بتوان این کلاس را از تغییرات بوجود آمده آگاه ساخت. روشی که من برای این کار درنظر گرفتم استفاده از یک اینترفیس حاوی اعضای موردنیاز برای پیاده‌سازی این امکان است تا مدیریت این ویژگی در ادامه راحت‌تر شود.

.

اینترفیس IDbCachedResourceManager

این اینترفیس به صورت زیر تعریف شده است:

```
namespace DbResourceProvider
{
  internal interface IDbCachedResourceManager
  {
    string ResourceName { get; }

    void ClearAll();
    void Clear(string culture);
    void Clear(string culture, string resourceKey);
  }
}
```

در پراپرتی فقط خواندنی ResourceName نام منبع کش شده ذخیره خواهد شد.

متد ClearAll برای پاک‌سازی تمامی ورودی‌های کش‌شده استفاده می‌شود.

متدهای Clear برای پاک‌سازی ورودی‌های کش‌شده یک کالچر به خصوص و یا یک ورودی خاص استفاده می‌شود.

با استفاده از این اینترفیس، پیاده‌سازی کلاس DbResourceManager به صورت زیر تغییر می‌کند:

```
using System.Collections.Generic;
using System.Globalization;
using DbResourceProvider.Data;
namespace DbResourceProvider
{
  internal class DbResourceManager : IDbCachedResourceManager
  {
    private readonly string _resourceName;
    private readonly Dictionary<string, Dictionary<string, object>> _resourceCacheByCulture;
    public DbResourceManager(string resourceName)
    {
      _resourceName = resourceName;
      _resourceCacheByCulture = new Dictionary<string, Dictionary<string, object>>();
    }
    public object GetObject(string resourceKey, CultureInfo culture) { ... }
    private object GetCachedObject(string resourceKey, string cultureName) { ... }

    #region Implementation of IDbCachedResourceManager
    public string ResourceName
    {
      get { return _resourceName; }
    }
    public void ClearAll()
    {
      lock (this)
      {
        _resourceCacheByCulture.Clear();
      }
    }
    public void Clear(string culture)
```

```
    {
      lock (this)
      {
        if (!_resourceCacheByCulture.ContainsKey(culture)) return;
        _resourceCacheByCulture[culture].Clear();
      }
    }
    public void Clear(string culture, string resourceKey)
    {
      lock (this)
      {
        if (!_resourceCacheByCulture.ContainsKey(culture)) return;
        _resourceCacheByCulture[culture].Remove(resourceKey);
      }
    }
    #endregion
  }
}
```

اعضای اینترفیس IDbCachedResourceManager به صورت مناسبی در کد بالا پیاده‌سازی شدند. در تمام این پیاده‌سازی‌ها مقادیر مربوطه از درون متغیر کشینگ پاک می‌شوند تا پس از اولین درخواست، بلافاصله از دیتابیس خوانده شوند. برای جلوگیری از دسترسی هم‌زمان نیز از بلاک lock استفاده شده است.

برای استفاده از این امکانات جدید همان‌طور که در بالا نیز اشاره شد باید بتوان نمونه‌های تولیدی از کلاس DbResourceManager توسط ASP.NET درون متغیری استاتیک ذخیره شوند. برای اینکار از کلاس جدیدی با عنوان DbResourceCacheManager استفاده می‌شود که برخلاف تمام کلاس‌های تعریف‌شده تا اینجا با سطح دسترسی public تعریف می‌شود.

.

کلاس DbResourceCacheManager

مدیریت نمونه‌های تولیدی از کلاس DbResourceManager در این کلاس انجام می‌شود. این کلاس پیاده‌سازی ساده‌ای به‌صورت زیر دارد:

```
using System.Collections.Generic;
using System.Linq;
namespace DbResourceProvider
{
  public static class DbResourceCacheManager
  {
    internal static List<IDbCachedResourceManager> ResourceManagers { get; private set; }
    static DbResourceCacheManager()
    {
      ResourceManagers = new List<IDbCachedResourceManager>();
    }
    public static void ClearAll()
    {
      ResourceManagers.ForEach(r => r.ClearAll());
    }
    public static void Clear(string resourceName)
    {
      GetResouceManagers(resourceName).ForEach(r => r.ClearAll());
    }
    public static void Clear(string resourceName, string culture)
    {
      GetResouceManagers(resourceName).ForEach(r => r.Clear(culture));
    }
    public static void Clear(string resourceName, string culture, string resourceKey)
    {
      GetResouceManagers(resourceName).ForEach(r => r.Clear(culture, resourceKey));
    }

    private static List<IDbCachedResourceManager> GetResouceManagers(string resourceName)
    {
      return ResourceManagers.Where(r => r.ResourceName.ToLower() == resourceName.ToLower()).ToList();
    }
```

```
  }
}
```

ازآنجاکه نیازی به تولید نمونه ای از این کلاس وجود ندارد، این کلاس به صورت استاتیک تعریف شده است. بنابراین تمام اعضای درون آن نیز استاتیک هستند.

از پراپرتی ResourceManagers برای نگهداری لیستی از نمونه‌های تولیدی از کلاس DbResourceManager استفاده می‌شود. این پراپرتی از نوع List<IDbCachedResourceManager> تعریف شده است و برای جلوگیری از دسترسی بیرونی، سطح دسترسی آن internal درنظر گرفته شده است.

در کانستراکتور استاتیک این کلاس (اطلاعات بیشتر درباره static constructor در اینجا) این پراپرتی با مقداردهی به یک نمونه تازه از لیست، اصطلاحا initialize می‌شود.

سایر متدها نیز برای فراخوانی متدهای موجود در اینترفیس IDbCachedResourceManager پیاده‌سازی شده‌اند. تمامی این متدها دارای سطح دسترسی public هستند. همان‌طور که می‌بینید از خاصیت ResourceName برای مشخص‌کردن نمونه موردنظر استفاده شده است که دلیل آن در قسمت قبل شرح داده شده است.

دقت کنید که برای اطمینان از انتخاب درست همه موارد موجود در شرط انتخاب نمونه موردنظر در متد GetResouceManagers از متد ToLower برای هر دو سمت شرط استفاده شده است.

.

نکته مهم: درباره علت برگشت یک لیست از متد انتخاب نمونه موردنظر از کلاس DbResourceManager در کد بالا (یعنی متد GetResouceManagers) باید نکته‌ای اشاره شود. در قسمت قبل عنوان شد که سیستم مدیریت منابع ASP.NET نمونه‌های تولیدی از پرووایدرهای منابع را به ازای هر منبع کش می‌کند. اما یک نکته بسیار مهم که باید به آن توجه کرد این است که این کش برای «عبارات بومی‌سازی **ضمنی** » و نیز «متد مربوط به **منابع محلی** » موجود در کلاس HttpContext و یا نمونه مشابه آن در کلاس TemplateControl (همان متد GetLocalResourceObject که درباره این متدها در قسمت سوم این سری شرح داده شده است) از یکدیگر جدا هستند و استفاده از هریک از این دو روش موجب تولید یک نمونه مجزا از پرووایدر مربوطه می‌شود که متاسفانه کنترل آن از دست برنامه نویس خارج است. دقت کنید که این اتفاق برای منابع کلی رخ نمی‌دهد.

بنابراین برای پاک کردن مناسب ورودی‌های کش‌شده در کلاس فوق به جای استفاده از متد Single در انتخاب نمونه موردنظر از کلاس DbResourceManager (در متد GetResouceManagers) از متد Where استفاده شده و یک لیست برگشت داده می‌شود. چون با توجه به توضیح بالا امکان وجود دو نمونه DbResourceManager از یک منبع درخواستی **محلی** در لیست نمونه‌های نگهداری شده در این کلاس وجود دارد.

.

افزودن نمونه‌ها به کلاس DbResourceCacheManager

برای نگهداری نمونه‌های تولید شده از DbResourceManager، باید در یک قسمت مناسب این نمونه‌ها را به لیست مربوطه در کلاس DbResourceCacheManager اضافه کرد. بهترین مکان برای انجام این عمل در کلاس پایه BaseDbResourceProvider است که درخواست تولید نمونه را در متد EnsureResourceManager درصورت نال بودن آن می‌دهد. بنابراین این متد را به صورت زیر تغییر می‌دهیم:

```
private void EnsureResourceManager()
{
  if (_resourceManager != null) return;
  {
    _resourceManager = CreateResourceManager();
    DbResourceCacheManager.ResourceManagers.Add(_resourceManager);
  }
}
```

تا اینجا کار پیاده‌سازی امکان مدیریت مقادیر کش‌شده در کتابخانه تولیدی به پایان رسیده است.

.

استفاده از کلاس DbResourceCacheManager

پس از پیاده‌سازی تمامی موارد لازم، حالتی را درنظر بگیرید که مقادیر ورودی‌های تعریف شده در منبع "dir1/page1.aspx" تغییر کرده است. بنابراین برای بروزرسانی مقادیر کش‌شده کافی است تا از کدی مثل کد زیر استفاده شود:

```
DbResourceCacheManager.Clear("dir1/page1.aspx");
```

کد بالا کل ورودی‌های کش‌شده برای منبع "dir1/page1.aspx" را پاک می‌کند. برای پاک کردن کالچر یا یک ورودی خاص نیز می‌توان از کدهایی مشابه زیر استفاده کرد:

```
DbResourceCacheManager.Clear("Default.aspx", "en-US");
DbResourceCacheManager.Clear("GlobalTexts", "en-US", "Yes");
```

.

دریافت کد پروژه

کد کامل پروژه DbResourceProvider به همراه مثال و اسکریپت‌های دیتابیسی مربوطه از لینک زیر قابل دریافت است:

DbResourceProvider.rar

برای استفاده از این مثال ابتدا باید کتابخانه Entity Framework (با نام EntityFramework.dll) را مثلا از طریق نوگت دریافت کنید. نسخه‌ای که من در این مثال استفاده کردم نسخه 4.4 با حجم حدود 1 مگابایت است.

نکته: در این کد یک بهبود جزئی اما مهم در کلاس ResourceData اعمال شده است. در قسمت سوم این سری، اشاره شد که نام ورودی‌های منابع Case Sensitive **نیست** . بنابراین برای پیاده‌سازی این ویژگی، متدهای این کلاس باید به صورت زیر تغییر کنند:

```
public Resource GetResource(string resourceKey, string culture)
{
  using (var data = new TestContext())
  {
    return data.Resources.SingleOrDefault(r => r.Name.ToLower() == _resourceName.ToLower() &&
r.Key.ToLower() == resourceKey.ToLower() && r.Culture == culture);
  }
}
```

```
public List<Resource> GetResources(string culture)
{
  using (var data = new TestContext())
  {
    return data.Resources.Where(r => r.Name.ToLower() == _resourceName.ToLower() && r.Culture ==
culture).ToList();
  }
}
```

تغییرات اعمال شده همان استفاده از متد ToLower در دو طرف شرط مربوط به نام منابع و کلید ورودی‌هاست.

.

در آینده...

در ادامه مطالب، بحث تهیه پرووایدر سفارشی فایلهای resx. برای پیاده‌سازی امکان به‌روزرسانی در زمان اجرا ارائه خواهد شد. بعد از پایان تهیه این پرووایدر سفارشی، این سری مطالب با ارائه نکات استفاده از این پرووایدرها در ASP.NET MVC پایان خواهد یافت.

.

منابع

http://msdn.microsoft.com/en-us/library/aa905797.aspx

http://www.west-wind.com/presentations/wwdbresourceprovider

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۳/۱۲ ۱۴:۲۳

با تشکر از زحمات شما.

یک بهبود جزئی: مطابق Managed Threading Best Practices بهتره از lock this استفاده نشه و از یک شیء object خصوصی استفاده شود.

.Use caution when locking on instances, for example lock(this) in C# or SyncLock(Me) in Visual Basic
.If other code in your application, external to the type, takes a lock on the object, deadlocks could occur

نویسنده: یوسف نژاد
تاریخ: ۱۳۹۲/۰۳/۱۲ ۱۴:۳۶

مطلب شما کاملا صحیح است.
ممنون بابت یادآوری.

نویسنده: علیرضا همتی
تاریخ: ۱۳۹۲/۰۳/۲۹ ۲۳:۵۷

سلام و تشکر از زحمات شما.
من نتوانستم از این پروایدر در displayAttribute و بقیه اتریبیوتها استفاده کنم. لطفا من و راهنمایی کنید.

نویسنده: یوسف نژاد
تاریخ: ۱۳۹۲/۰۳/۳۰ ۱۰:۳۰

متاسفانه امکان استفاده مستقیم از این پرووایدرهای سفارشی در این attributeها در MVC میسر نیست. این attributeها به جای استفاده از پرووایدر منابع برای استخراج مقادیر ورودی‌ها طوری طراحی شده اند که با استفاده از Reflection از داده‌های ارائه شده مقادیر را از کلاس و پراپرتی مربوطه استخراج کنند. بنابراین در این attributeها نمیتوان جایی برای استفاده از پرووایدرهای منابع یافت.

برای حل این مشکل چندین راه حل وجود دارد:
مثلا attributeهای موردنیاز توسط خود برنامه نویس پیاده‌سازی شوند.
یا اینکه یک کلاس مخصوص ایجاد کرد و استخراج مقادیر ورودی‌های منابع را در آن پیاده‌سازی کرد و در attributeهای موردنیاز از نام این کلاس و پراپرتی‌های درون آن استفاده کرد.
یا اگر از فایلهای resx. استفاده می‌شود یک ابزار سفارشی برای تولید کلاس مرتبط با منبع اصلی مثل ابزار توکار ویژوال استودیو (PublicResXFileCodeGenerator) تولید کرد تا کلاس‌های تولیدی به جای استفاده از ResourceManager از پرووایدر منابع استفاده کند (با استفاده از متدهای موجود در HttpContext).
البته این روش‌ها برای حل مشکلات مربوطه در MVC در ادامه این سری شرح داده می‌شوند.

نویسنده: علیرضا همتی
تاریخ: ۱۳۹۲/۰۳/۳۰ ۱۳:۱۱

ممنون از شما.

نویسنده: محسن موسوی
تاریخ: ۱۳۹۲/۰۶/۰۳ ۱۷:۳۹

با تشکر از زحمات شما
اینجا بیان شده زمانیکه از اسمبلی دیگری برای resourceها استفاده میکنید فقط میتوان **global resources** را پوشش داد.
بنابراین برای استفاده از کلاس LocalDbResourceProvider بایستی تغییراتی صورت بگیره.
چونکه همیشه این متد

```
using System.Web.Compilation;

namespace DbResourceProvider
{
  internal class DbResourceProviderFactory : ResourceProviderFactory
  {
    #region Overrides of ResourceProviderFactory

    public override IResourceProvider CreateGlobalResourceProvider(string classKey)
    {
      return new GlobalDbResourceProvider(classKey);
    }

    ...
  }
}
```

اجرا میشود.

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۰۷/۰۸ ۹:۵۸

مهندس عزیز با تشکر از کار گرانقدر شما
یک سوال؟
چگونه می‌توان الگوی کار را در این پروایدر گنجاند؟
آیا اصلا چنین امکانی دارد یا خیر؟

عنوان: **بررسی تغییرات ASP.NET MVC 5 beta1**
نویسنده: مهدی سعیدی فر
تاریخ: ۱۳۹۲/۰۴/۱۰ ۲۳:۱۵
آدرس: www.dotnettips.info
گروه‌ها: ASP.NET MVC 5, MVC

همانطور که می‌دانید، مایکروسافت در کنفرانس Build 2013 که چند روز پیش برگزار شد، Visual Studio 2013 Preview را به همراه ASP.NET MVC 5 beta1 و Entity Framework 6 beta 1 و تعدادی محصول دیگر، معرفی کرد.
در طی این مقاله قصد دارم تجربیات کار خودم با نسخه‌ی پیش نمایش MVC 5 را به اشتراک بزارم و نه صرفا بررسی یک change-log ساده.
برای کار با MVC 5 شما ابتدا باید یکی از نسخه‌های Visual Studio 2013 را نصب کنید. من در مقاله از Visual Studio Express 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 اند.

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.

همه‌ی 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.

Solution 'WebApplication5' (1 project)
WebApplication5
Properties
References
App_Data
App_Start
BundleConfig.cs
FilterConfig.cs
IdentityConfig.cs
RouteConfig.cs
Startup.Auth.cs
Content
themes
bootstrap-responsive.css
bootstrap-responsive.min.css
bootstrap.css
bootstrap.min.css
Site.css
Controllers
img
Models
Scripts
_references.js
bootstrap.js
bootstrap.min.js
jquery-1.8.2.intellisense.js
jquery-1.8.2.js
jquery-1.8.2.min.js
jquery-ui-1.8.24.js
jquery-ui-1.8.24.min.js
jquery.unobtrusive-ajax.js
jquery.unobtrusive-ajax.min.js
jquery.validate-vsdoc.js
jquery.validate.js
jquery.validate.min.js
jquery.validate.unobtrusive.js
jquery.validate.unobtrusive.min.js
modernizr-2.6.2.js
Views
favicon.ico
Global.asax
packages.config
Startup.cs
Web.config

همان طور که می‌بینید ساختار اصلی با نسخه‌های قبلی هیچ تفاوتی نکرده و تنها کتاب خانه ای 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 **bootstrap** و **responsive** بودن آن، خودنمایی می‌کنند.

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.
Web Hosting
You can easily find a web hosting company that offers the right mix of features and price for your applications.

Home Page - My ASP.NET
localhost:5820
Application name
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

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,

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** ، Entity Framework 6 و .NET 4.5 ، 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

Entity Framework نسخه‌ی 6 beta1 به صورت پیش فرض در پروژه وجود دارد. **خوشبختانه دیگر خبری هم از System.Data.Entity نیست.** همچنین حضور پررنگ **Owin** و **ASP.NET Identity** را متوجه خواهید شد.

خب قبلا 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 کاملا از نو نوشته شده، در بررسی‌های اولیه من، کدهای تولیدی چندان تفاوتی با نسخه‌ی قبل نداشت.احتمالا تغییرات در جای دیگری است.

حال اگر بخواهیم برای Controller ایجاد شده یه View ایجاد کنیم باید طبق روال سابق از شرتکات 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 فولدر Views شده و به صورت دستی فولدر جدیدی ایجاد و از گزینه‌ی Scaffold برای افزودن View جدید اقدام کرد.

```
1  @{
2      ViewBag.Title = "Index";
3  }
4  <div ng->
5      <
6  </div
7
```

ng-animate
ng-app
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 استادیو به صورت توکار، از **AngularJs** پشتیبانی می‌کند.

نتیجه گیری:

گویا مایکروسافت نیز به این نتیجه رسیده که ASP.NET MVC در نسخه‌ی سوم خود، کاملا 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 bootstrap و Angularjs شاهد هستیم. گویا خط مشی تیم توسعه دهنده نیز این گونه است. از جمله تغییرات خوب بحث Identity هست که کاملا 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: ۱۳۹۲/۰۴/۱۱ ۰:۲۷

- نکات مهم Bootstrap رو ما در سایت جاری بررسی کردیم و الزاما برای استفاده از آن نیازی به MVC5 نیست. همین الان در MVC4 هم می‌تونید ازش استفاده کنید. ولی درکل هر وقت مایکروسافت دست روی چیزی می‌گذارد، مزیتش تهیه حداقل 20 جلد کتاب جدید در مورد CSS و Bootstrap 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 همین Owin ایی 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 VS 2013 داشت.
- همچنین با توجه به سورس باز بودن MVC، دنبال کردن History سورس 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: ۱۳۹۲/۰۴/۲۳ ۹:۲۵

دمویی از تغییرات انجام شده در MVC5 و Web API 2

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: ۱۳۹۲/۰۹/۱۰ 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 mvc 4 من از وب ماتریکس WebMatrix.WebData.WebSecurity.CurrentUserName استفاده میکردم، ولی الان با mvc 5 نال میشه و مقدار نداره.

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۹/۱۰ ۱۸:۳

روش web matrix آنچنان مقبولیتی پیدا نکرده (^). بهتره از روش پروژه سورس باز IRIS استفاده کنید که مبتنی است بر Forms Authentication و مباحث Roles را هم داره.

نویسنده: منصور جعفری
تاریخ: ۱۳۹۲/۱۲/۲۹ ۱۲:۷

سلام

چطور میشه مثل MVC4 یک پروژه خالی از MVC 5 به وجود آورد و یا به‌طور کل آیا خود ویژوال استادیو MVC 5 رو بصورت پیش‌فرض داخل خودش نداره ؟
یعنی به این‌شکل که MVC 4 وجود داره MVC5 به ویژوال استادیو اضافه بشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۲۹ ۱۲:۵۱

در VS 2013 به همراه آخرین به روز رسانی‌ها:

- انتخاب دات نت 4.5 یا 4.5.1 به معنای کار با MVC 5.x است:

سپس در صفحه‌ی ظاهر شده، امکان انتخاب گزینه‌ی خالی نیز هست:

New ASP.NET Project - MV
Select a template:
Empty
Web Forms
MVC
Web API
Single Page Application
Facebook
Add folders and core references for:
Web Forms
MVC
Web API
Add unit tests
Test project name:
MVC5Empty.Tests

عنوان: Attribute Routing در ASP.NET MVC 5
نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۰/۲۵ ۲:۴۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, ASP.Net MVC, ASP.NET MVC 5

Routing مکانیزم مسیریابی ASP.NET MVC است، که یک URI را به یک اکشن متد نگاشت می‌کند. MVC 5 نوع جدیدی از مسیر یابی را پشتیبانی میکند که Attribute Routing یا مسیریابی نشانه ای نام دارد. همانطور که از نامش پیداست، مسیریابی نشانه ای از Attributeها برای این امر استفاده میکند. این روش به شما کنترل بیشتری روی URIهای اپلیکیشن تان می‌دهد.
مدل قبلی مسیریابی (conventional-routing) هنوز کاملا 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 نشانه ای را بررسی میکنیم.
چرا مسیریابی نشانه ای؟
فعال سازی مسیریابی نشانه ای
پارامترهای اختیاری URI و مقادیر پیش فرض
پیشوند مسیر ها
مسیر پیش فرض
محدودیت‌های مسیر ها محدودیت‌های سفارشی

نام مسیر ها
ناحیه‌ها (Areas)

چرا مسیریابی نشانه ای

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 کنترلرها به 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 اکشن متدها تعریف می‌شوند، یعنی در یک فایل سورس و نه در یک کلاس پیکربندی خارجی، درک و فهم نگاشت URIها به اکشن‌ها واضح‌تر و راحت می‌شود. تعریف مسیر قبلی، می‌تواند توسط یک attribute ساده بدین صورت نگاشت شود:

```
[Route("{productId:int}/{productTitle}")]
public ActionResult Show(int productId) { ... }
```

فعال سازی Attribute Routing

برای فعال سازی مسیریابی نشانه ای، متد MapMvcAttributeRoutes را هنگام پیکربندی فراخوانی کنید.

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

        routes.MapMvcAttributeRoutes();
    }
}
```

همچنین می‌توانید 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 می‌کنید.

```
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 متد "View" نگاشت می‌شوند، مسیر اول تمام کتاب‌ها را لیست میکند، و مسیر دوم جزئیات کتابی مشخص را لیست می‌کند. هر دو مسیر books/lang/ و books/lang/en/ به یک شکل نگاشت می‌شوند، 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 می‌شوند. بعنوان 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

می توانید خاصیت [Route] را روی کنترلر اعمال کنید، تا اکشن متد را بعنوان یک پارامتر بگیرید. این مسیر سپس روی تمام اکشن متدهای 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 constraintها یا محدودیت هایی که پشتیبانی می‌شوند را لیست می‌کند.

مثال	توضیحات	محدودیت
{x:alpha}	کاراکترهای الفبای لاتین را تطبیق (match) می‌دهد (a-z, A-Z).	alpha
{x:bool}	یک مقدار منطقی را تطبیق می‌دهد.	bool
{x:datetime}	یک مقدار **DateTime** را تطبیق می‌دهد.	datetime
{x:decimal}	یک مقدار پولی را تطبیق می‌دهد.	decimal
{x:double}	یک مقدار اعشاری 64 بیتی را تطبیق می‌دهد.	double
{x:float}	یک مقدار اعشاری 32 بیتی را تطبیق می‌دهد.	float
{x:guid}	یک مقدار GUID را تطبیق می‌دهد.	guid
{x:int}	یک مقدار 32 بیتی integer را تطبیق می‌دهد.	int
{x:length(6)} {x:length(1,20)}	رشته ای با طول تعیین شده را تطبیق می‌دهد.	length
{x:long}	یک مقدار 64 بیتی integer را تطبیق می‌دهد.	long
{(x:max(10}	یک مقدار integer با حداکثر مجاز را تطبیق می‌دهد.	max
{(x:maxlength(10}	رشته ای با حداکثر طول تعیین شده را تطبیق می‌دهد.	maxlength
{(x:min(10}	مقداری integer با حداقل مقدار تعیین شده را تطبیق می‌دهد.	min
{(x:minlength(10}	رشته ای با حداقل طول تعیین شده را تطبیق می‌دهد.	minlength
{x:range(10,50)}	مقداری integer در بازه تعریف شده را تطبیق می‌دهد.	range
{($${x:regex(^\d{3}-\d{3}-\d{4}	یک عبارت با قاعده را تطبیق می‌دهد.	regex

توجه کنید که بعضی از constraint ها، مانند "min" آرگومان‌ها را در پرانتز دریافت می‌کنند.

می توانید محدودیت‌های متعددی روی یک پارامتر تعریف کنید، که باید با دونقطه جدا شوند. بعنوان 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/{message:maxlength(3)?}")]
public ActionResult Greet(string message) { ... }
```

محدودیت‌های سفارشی

با پیاده سازی قرارداد IRouteConstraint می‌توانید محدودیت‌های سفارشی بسازید. بعنوان مثال، constraint 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 رجیستر کردن این constraint 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. بعنوان 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>
```

ناحیه‌ها (Areas)

برای مشخص کردن ناحیه ای 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. بعنوان مثال:

```
[RouteArea("BackOffice", AreaPrefix = "back-office")]
```

اگر از ناحیه‌ها هم بصورت مسیریابی نشانه ای، و هم بصورت متداول (که با کلاس‌های AreaRegistration پیکربندی می‌شوند) استفاده می‌کنید باید مطمئن شوید که رجیستر کردن نواحی اپلیکیشن پس از مسیریابی نشانه ای پیکربندی می‌شود. به هر حال رجیستر 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 registration و در آخر default route رجیستر شوند. بعنوان مثال:

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

```
    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: ۱۳۹۲/۱۲/۲۹ ۰:۲۷

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: ۱۳۹۲/۱۲/۲۹ ۱:۱۱

برای ریشه سایت ویژگی زیر را هم باید اضافه کنید:

```
[Route("~/")]
```

نویسنده: منصور جعفری
تاریخ: ۱۳۹۲/۱۲/۲۹ ۱۲:۳۳

ممنونم که پاسخ دادید یعنی باید به این شکل بنویسم

```
[Route("~/Page/{page?}")]
        public ActionResult List(int page=1)
        {
```

نیازی به تعریف اکشن یا RoutePrifix نیست؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۲۹ ۱۲:۴۵

قابلیت ترکیب دارند:

```
[Route("~/")]
[Route("Page/{page?}")]
public ActionResult List(int page=1)
{
```

نویسنده: یاشار راشدی
تاریخ: ۱۳۹۳/۰۳/۲۹ ۱۳:۲۲

در صورتی که از Areaها استفاده کنید باید بالای هر کنترلر داخل Area حتما Prefix مربوط به Area را اضافه کنید وگرنه Exception دریافت می‌کنید.

```
[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

عنوان: **قابلیت Attribute Routing در ASP.NET MVC 5**
نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۲/۰۹/۱۷ ۸:۵
آدرس: www.dotnettips.info
گروه‌ها: ASP.NET MVC 5, routing, Attibute Routing, MVC

در ASP.NET MVC 5 یک قابلیت جدید با نام Attribute Routing افزوده شده است که به ما این اجازه را می‌دهد تا Routeهای 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 :

ابتدا 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 می‌گویم که تمام اکشن متدهای 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 متد Index را با افزودن ویژگی Route که آدرس products/~ را تطبیق می‌دهد تعیین کرده ایم.

نحوه تعیین Optional URI Parameter :
کافی است علامت سوال را به آخر پارامتر اضافه کنیم :

```
[Route("{id?}")]
public ActionResult Index(int id)
{
   return View();
}
```

نحوه تعیین Default Route :

```
[RoutePrefix("products")]
[Route("{action=index}")]
public class ProductsController : Controller
{
    public ProductsController()
    {
    }
    public ActionResult Index()
    {
```

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

نحوه تعیین Constraint برای Routeها :

```
[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 نمی‌دهد.
همچنین می‌توانید 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 نمی‌گیرد:

```
[Route("{title:regex(\u0600-\u06FF)}")]
public ActionResult Search(string title)
{
   return View();
}
```

در لینکی که در بالا معرفی شده لیست کامل Constraintها را می‌توانید مشاهده نمائید،

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 مثال 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 MVC 4 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 مستقیما به اکشن Index فوروارد بشه یعنی : http://localhost/products/10

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 attribute routing 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 attribute routing و convention-based routing با هم وجود دارد و اگر در حالت قدیمی یعنی convention-based routing، شما مثل قبل یک default route تعریف کرده باشید، دیگر نیازی به ذکر و تکرار default route هم نام یک کنترلر در حالت attribute routing نخواهید داشت و فقط جایی که واقعا نیاز است باید از آن استفاده کرد. attribute routing کار رو برای تعریف 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 سئوال داشتم: برای ایجاد زیر دامنه چه راهکاری هست؟
مثلا: forum.site.com

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۶/۱۲ ۱۲:۳۴

ساب دومین ارتباطی به مسیریابی نداره چون مسیریابی مرتبط 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 routes.MapMvcAttributeRoutes ، RegisterRoutes(); هم اضافه کردم ممنون میشم راهنماییم کنید

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۶/۱۳ ۱۱:۳۱

دیباگش کنید تا مشخص شه کدوم مسیریابی اعمال شده: چطور مسیریابی‌های ASP.NET MVC را دیباگ کنیم؟

عنوان: **یک تکنیک جالب در نحوه نام گذاری فیلدهای دیتابیس به منظور استفاده بهینه از فایل های T4 در MVC5**
نویسنده: صادق نجاتی
تاریخ: ۱۳۹۳/۰۲/۱۰ ۱۰:۴۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, Kendo UI, T4

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. مثلا 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. مجددا 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 شده‌است 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 وسیع‌تر است. مثلا حالتی را در نظر بگیرید که می‌خواهیم از طریق Scaffolding برای یک جدول بانک اطلاعاتی که یک فیلد آن آدرس یک تصویر را نگهداری می‌کند View ایجاد نماییم. خوب ما در صفحه Index می‌خواهیم تصویر مورد نظر با اندازه 100 * 100 پیکسل نمایش دهیم (چون قرار است لیستی از تصاویر نمایش داده شود باید در اندازه قابل نمایشی باشد) ولی 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 ، فایل‌های T4 را (List.t4 و Details.t4) سفارشی سازی نماییم.

نویسنده: محمد جلیل عبدالله زاده
تاریخ: ۱۳۹۳/۱۲/۱۷ ۱:۳

دوستان ممکنه منو راهنمایی کنید چطور میتونم به CustomAttributesهای یک Property تو این ساختار دسترسی داشته باشم

با توجه به اینکه نوع Property در اینجا PropertyMetaData هستش ولی تو MVC4 و قبل از اون ProprtyInfo بوده و چنین چیزی امکان پذیر بوده

نویسنده: محسن موسوی
تاریخ: ۱۳۹۳/۱۲/۱۷ ۱۰:۳۷

میتونی از AdditionalMetadata استفاده کنی.
اطلاعات بیشتر

عنوان: **نگاهی به هویت سنجی کاربران در ASP.NET MVC 5**
نویسنده: مرتضی دلیل
تاریخ: ۱۳۹۳/۰۳/۲۲ ۱۱:۵
آدرس: www.dotnettips.info
گروه‌ها: Entity framework, MVC, Security, ASP.NET Identity

در مقاله پیش رو، سعی شده‌است به شکلی تقریبا 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 MVC 4 برای دسترسی به جداول مرتبط با اعتبار سنجی (مثلا 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 پروژه ای بسازید و حالت پیش فرض آن را تغییر ندهید و آن را اجرا کنید و ثبت نام را انجام دهید، بلافاصله 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 IUser و IRole 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 آن‌چیزی را دارد که در پکیج اول معرفی شده است. کلاس‌های 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 واسط (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 مدل‌های مربوط به برنامه خود را ساختیم، اولا نیاز به ساخت کلاس کانتکست نداریم چون خود MVC5 کلاس کانتکست را دارد؛ ثانیا نیاز به ایجاد مدل برای جداول اعتبارسنجی نیست، چون کلاسی برای فیلدهای 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 کلاسهایش Context است). همانطور 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 پراپرتی اضافه کنیم، بطور اتوماتیک آن پراپرتی به جدول ASPNetUsers در دیتابیس اضافه می‌شود و دیگر نگران فیلدهای نداشته‌ی جدول کاربران ASP.NET نخواهیم بود. مثلا در کد زیر چند عنوان به این جدول اضافه کرده ایم.

```
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 پراپرتی به جدول کاربران اضافه میشود. دقت داشته باشید با اینکه هیچ مدلی برای جدول کاربران نساخته ایم اما کلاس ApplicatioUsers کلاسی است که به ما امکان دسترسی به مقادیر این جدول را می‌دهد(دسترسی به معنای اضافه و حذف وتغییر مقادیر این جدول است) (در MVC4 به کمک کلاس membership کارهای مشابهی انجام میدادیم)

در ساختن مدل هایمان نیز اگر نیاز به ارتباط با جدول کاربران باشد، از همین کلاس فوق استفاده میکنیم. کلاس واسط(مدل واسط) بین AspNetUsers و Product در کد زیر 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. پراپرتی 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 = "لطفا {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 = "لطفا {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 ریال}")]
        [UIHint("Integer")]
        [Required(ErrorMessage = "لطفا {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, ApplicationUserId = "2" },
              new AuthorProduct() { AuthorProductId = 2, ProductID = 2, ApplicationUserId = "1" },
              new AuthorProduct() { AuthorProductId = 3, ProductID = 3, ApplicationUserId = "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. سازنده این کلاس به کانتکست 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 خوبخود 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.

عنوان: **سفارشی کردن ASP.NET Identity در MVC 5**
نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۰/۱۹ ۱۹:۵۰
آدرس: www.dotnettips.info
گروه‌ها: Entity framework, MemberShip, ASP.NET MVC 5, ASP.NET Identity, MVC, Security

یکی از نیازهای رایج توسعه دهندگان هنگام استفاده از سیستم عضویت ASP.NET سفارشی کردن الگوی داده‌ها است. مثلا 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 *UserProfile* در فریم ورک SimpleMembership است. این کلاس خالی 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 بر خلاف SimpleMembership یک عدد صحیح یا int نیست، بلکه بصورت یک رشته ذخیره 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 می‌کنیم 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 می‌دهید 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 می‌شود، روی الگوی دیتابیس سیستم عضویت کنترل کامل دارید. بنابراین به سادگی می‌توانید با استفاده از قابلیت Code First مدل‌های خود را سفارشی کنید.

در پست‌های آتی این مطلب را ادامه خواهیم داد تا ببینیم چگونه می‌توان ایمیل‌های 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 بدین و از یک DbContext استفاده کنید. یعنی DbSetهای 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 وب‌فرمز استفاده میکنم از این امکان. فایل IdentityModels.cs رو نداره توی پروژه .
مشکل کجاس؟
-enable migrations هم زدم خطا داد

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۱/۱۱ ۱:۵۳

از VS 2013 استفاده می‌کنی ؟

نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۱/۱۱ ۲:۵۱

لازم نیست تمام این آبجکت‌ها رو به context نگاشت کنید. قالب پروژه‌های VS 2013 بصورت خودکار در پوشه Models کلاسی بنام IdentityModels میسازه. این کلاس شامل کلاسی بنام ApplicationDbContext میشه که تعریفی مانند لیست زیر داره:

```
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 IdentityUser ارث بری کنه، حال با نام پیش فرض یا با نام دلخواه. سپس باید کلاسی بسازید که از UserManager<T> مشتق میشه. با استفاده از این کلاس می‌تونید به موجودیت‌های کاربران دسترسی داشته باشید. بعنوان 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 می‌کنیم. 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 ASP.NET Identity با مدل EF Code-First کار میکنه.

نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۱/۱۱ ۲:۵۷

از VS 2013 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 می‌تونید ایجاد کنید. اگر به نظرات بالا مراجعه کنید گفته شد که کلاس کاربران باید از

IdentityModels ارث بری کنه.

نویسنده: رجایی
تاریخ: ۱۳۹۲/۱۱/۱۲ ۹:۲۱

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 ولی باز هم این ایراد رو در اتصال به دیتابیس نشان می‌دهد. من از add-migration هم استفاده می‌کنم تا تغییرات موجودیت‌ها رو کامل به من نشان دهد که چیزی را عنوان نمی‌کند.

نویسنده: افتاب
تاریخ: ۱۳۹۲/۱۲/۲۷ ۲۳:۳۸

سلام و متشکر ،
دنبال آن میگشتم که چه طوری میشه دو تا صفحه لوگین در پروژه داشت که ورودی کاربران از مدیران جدا باشد

نویسنده: افتاب
تاریخ: ۱۳۹۲/۱۲/۲۷ ۲۳:۴۴

میشه در مورد «این کلاس رو کلا حذف کنید، چون قراره از یک DbContext برای تمام موجودیت‌ها استفاده کنید....» بیشتر توضیح بفرمایید؟

نویسنده: سعید رضایی
تاریخ: ۱۳۹۳/۰۲/۰۱ ۱۶:۴۱

با عرض سلام.تو Vs2012+mvc4 نمیشه از identity استفاده کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۰۱ ۱۷:۲۶

خیر. با دات نت 4.5 کامپایل شده.

نویسنده: میثم سلیمانی
تاریخ: ۱۳۹۳/۰۵/۲۶ ۱۱:۷

با سلام و احترام
من دستور زیر رو تو asp.net mvc Identity sample 2 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 doen'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 DataSetهای 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. ApplicationDbContext 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 ، در حقیقت Context اصلی کاری برنامه است.

نویسنده: سانی
تاریخ: ۱۳۹۳/۱۱/۲۵ ۱۲:۴۶

سلام

این سوال رو توی asp.net , barnamenevis هم پرسیدم. من برای Identity از sample خود MVC استفاده کردم؛ به این صورت که با استفاده از nuget آن را نصب کرم. حال توی یک لیست بازشو میخوام فقط کاربرانی را نشان دهم که یک نقش را دارند.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۱/۲۵ ۱۴:۵۱

مثال مطلب اعمال تزریق وابستگی‌ها به مثال رسمی ASP.NET Identity ، جهت تکمیل کلاس ApplicationRoleManager آن بهبود داده شد (^). برای نمونه :

```
public IList<ApplicationUser> GetApplicationUsersInRole(string roleName)
{
    var selectedUserIds = from role in this.Roles
                          where role.Name == roleName
                          from user in role.Users
                          select user.UserId;
    return _users.Where(applicationUser => selectedUserIds.Contains(applicationUser.Id)).ToList();
}
```

عنوان: **افزودن تصدیق ایمیل به ASP.NET Identity در MVC 5**
نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۰/۱۹ ۲۱:۴۵
آدرس: www.dotnettips.info
گروه‌ها: Entity framework, Tips, ASP.NET MVC 5, ASP.NET Identity, MVC, Security

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* حالا باید بدین شکل باشد.

```
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 گرفته ام. پس از ایجاد حساب کاربری باید شناسه تولید شده را به آن اضافه کنیم و مقدار فیلد IsConfirmed 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 (plain text) یا HTML بسازید، عکس و فایل در آن درج و ضمیمه کنید و امکانات بسیار خوب دیگر. اکشن متد RegisterStepTwo تنها کاربر 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 متد *RegisterConfirmation* باز می‌گردیم.

```
private bool ConfirmAccount(string confirmationToken)
{
    ApplicationDbContext context = new ApplicationDbContext();
    ApplicationUser user =  context.Users.SingleOrDefault(u => u.ConfirmationToken ==
confirmationToken);
    if (user != null)
    {
        user.IsConfirmed = true;
        DbSet<ApplicationUser> dbSet = context.Set<ApplicationUser>();
        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");
}
```

متد *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 IsConfirmed 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 متد *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.IsConfirmed)
        {
            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 خطایی 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 متد SendEmailConfirmation دقت کنید خواهید دید که آبجکتی از نوع Email می‌سازیم (که در اسمبلی‌های 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 Views پوشه 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 حاظر بصورت plain text ارسال می‌شوند. برای اطلاعات بیشتر درباره ایمیل‌های HTML و امکانات پیشرفته‌تر به سایت پروژه Postal مراجعه کنید.
همانطور که مشاهده می‌کنید در این نما همان خاصیت‌های دینامیک تعریف شده را فراخوانی می‌کنیم تا مقادیر لازم را بدست آوریم.

To. ViewBag آدرس ایمیل گیرنده را نشان می‌دهد.
UserName. ViewBag نام کاربر جاری را نمایش می‌دهد.
ConfirmationToken. ViewBag شناسه تولید شده برای تایید کاربر است.

در این قالب لینکی به متد 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: ۱۳۹۲/۱۰/۱۹ ۲۲:۱۴

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: ۱۳۹۲/۱۰/۱۹ ۲۳:۷

با تشکر از شما، پست بروز رسانی شد.

نویسنده: کاربر
تاریخ: ۱۳۹۲/۱۱/۰۵ ۱۶:۵۲

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 timed out میده

```
<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: ۱۳۹۲/۱۱/۰۵ ۱۹:۵۲

```
<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: ۱۳۹۲/۱۲/۰۵ ۰:۲۷

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: ۱۳۹۳/۰۲/۰۱ ۱۵:۱۳

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: ۱۳۹۳/۰۲/۰۱ ۱۷:۱۲

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. system.net 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 پوشه‌ی Views برنامه.

عنوان: **ایجاد کپچایی (captcha) سریع و ساده در ASP.NET MVC 5**
نویسنده: مرتضی دلیل
تاریخ: ۱۳۹۳/۰۳/۲۶ ۹:۲۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, Security, CAPTCHA

در این مثال به کمک MVC5 ، یک کپچای ساده و قابل فهم را تولید و استفاده خواهیم کرد. این نوشته بر اساس این مقاله 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 متد 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 3+4). این تصویر خروجی این اکشن متد است. به سادگی می‌توانید این اکشن را بر اساس خواسته خود اصلاح کنید؛ مثلا به جای حاصل جمع دو عدد، از کاربرد چند حرف یا عدد که بصورت اتفاقی تولید کرده‌اید، استفاده نمائید.

فرض کنید می‌خواهیم کپچا را هنگام ثبت نام استفاده کنیم.

در فایل AccountViewModels.cs در پوشه مدل‌ها در کلاس RegisterViewModel خاصیت زیر را اضافه کنید:

```
[Required(ErrorMessage = "لطفا {0} را وارد کنید")]
        [Display(Name = "حاصل 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)

                        <img alt="Captcha" id="imgcpatcha" src="@Url.Action("CaptchaImage","Captcha")"
style="" />
                    </div>
```

وظیفه این بخش، نمایش کپچاست . تگ img دارای آدرسی است که توسط اکشن متدی 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 را به شکل زیر بنویسید (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) {
                $("#imgcpatcha").attr("src", "/Captcha/CaptchaImage?" + functionResult);
            });

        });
    });
</script>
```

آنچه در url نوشته شده است، شاید اصولی‌ترین شکل فراخوانی یک اکشن متد 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 متد فرستاده نمیشود، اما اکشن متد 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 متد 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، بررسیِ برابریِ 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) . بدیهی‌است اگر این دو مقدار نابرابر باشند، یک خطا به ModelState اضافه می‌شود و شرط ModelState.IsValid که در اولین خط بعد از کد فوق وجود دارد، برقرار نخواهد بود و پیغام خطا 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 (بر خلاف کروم) تصاویر را نگهداری میکنند و آپدیت به روش فوق به مشکل برخورد میکند 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 است. پس نیازی 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: ۱۳۹۳/۰۳/۲۶ ۹:۳۳

ممنون 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. متد GetBuffer حجمی نزدیک به 2 برابر آرایه اصلی رو عموما داره و انتهاش 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.

نویسنده: ح مراداف
تاریخ: ۱۳۹۳/۱۱/۲۳ ۰:۸

با سلام و با تشکر؛ با اجازه بنده کد فوق رو کامل‌تر کردم و یک سری کد جدید بهش اضافه کردم و برخی بخش‌ها رو هم تغییر داده ام:

1- به جای سوال ، بنده یک عبارت رو نمایش میدم

2- ارسال دیتا از طریق کوئری استرینگ که باعث میشه سشن دیگه نیاز نباشه و از مصرف حافظه رو تا حد زیادی کاسته بشه. البته این مورد برای سایت‌های پربازدید خیلی قابل لمس است و ممکنه روی سایت‌های معمولی تفاوت زیادی احساس نشه.

3- ارسال داده بصورت هش شده ، که این رو بنده خودم با یک کلاس دست ساز معمولی به روش TripleDes انجام داده ام که دوستان به هر روشی می‌تونن داده هاشون رو هش کنن.

4- یکم حروف رو چرخوندم و فاصله بین حروف رو هم طوری تنظیم کردم که در عرض تصویر پخش بشن (از کل عرض تصویر استفاده بشه)

* شایان ذکر است که به نظر من روش فوق در ایجاد نویزهای دایره ای بسیار زیبا بود، چون همیشه همه جا با یک سری خط ساده نویز ایجاد می‌کنن ولی روش فوق واقعا خلاقانه و قشنگ بود :)

ساختار کنترلر ریکپچای من :

```
public class CaptchaController : Controller
    {
        private static readonly Brush ForeColor = Brushes.Black;
        private const string FontName = "tahoma";
        private const int FontSize = 14;
        private const int Width = 130;
        private const int Height = 35;

        [HttpGet]
        public ActionResult Image(string cc)
        {
            if (string.IsNullOrEmpty(cc) || string.IsNullOrWhiteSpace(cc))
                return null;

            var captchaData = CustomHashing.DecryptTpl(cc);

            var rand = new Random((int)DateTime.Now.Ticks);

            // image stream
            FileContentResult img = null;

            using (var mem = new MemoryStream())
            using (var bmp = new Bitmap(Width, Height))
            using (var mtrx = new Matrix())
            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
                int rn, xn, yn;
                var pen = new Pen(Color.Yellow);

                for (int i = 1; i < 10; i++)
                {
                    pen.Color = Color.FromArgb((rand.Next(0, 255)), (rand.Next(0, 255)), (rand.Next(0,
255)));

                    rn = rand.Next(0, (130 / 3));
                    xn = rand.Next(0, 130);
                    yn = rand.Next(0, 30);

                    gfx.DrawEllipse(pen, xn - rn, yn - rn, rn, rn);
                }

                //add chars
                #region draw pic

                float x = 1, y = 1;
                int degree = 10;

                for (int i = 0; i < captchaData.Length; i++)
                {
                    mtrx.Reset();

                    x = (float)(Width * (0.19 * i));

                    y = (float)(Height * 0.19);

                    degree = rand.Next(-25, 25);

                    if (i == 0 && degree > 20)
                    {
                        x += (FontSize + 5);
                        y -= 15;
                    }

                    mtrx.RotateAt(degree, new PointF(x, y));

                    gfx.Transform = mtrx;

                    gfx.DrawString(captchaData[i].ToString(), new Font(FontName, FontSize), ForeColor,
x, y);

                    gfx.ResetTransform();
                }
                #endregion

                //render as Jpeg
                bmp.Save(mem, System.Drawing.Imaging.ImageFormat.Jpeg);
                img = this.File(mem.GetBuffer(), "image/Jpeg");
            }

            return img;
        }
```

برای استفاده هم داریم :

```
@{
    var r = new Web.Tools.CustomRandom();
    string hash = Web.Tools.CustomHashing.EncryptTpl(r.CraeteCapchaNumericData(4));
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>test Index</title>
</head>
<body>
<div>

    <img src="@Url.Action("Image", "Captcha", new { cc = hash })" />
```

```
</div>
</body>
</html>
```

محتوای کلاس CustomRandom :

این کلاس به تعداد مورد نیاز کاراکتر عددی/عددی-حروفی می‌سازه و به شما تحویل میده

```
public class CustomRandom
 {
        /// <summary>
        /// ساخت یک عبارت عددی رندوم
        /// </summary>
        public string CraeteCapchaNumericData(int length)
        {
            var rnd = new Random((int) DateTime.Now.Ticks);
            var temp = new StringBuilder();

            for (var i = 0; i < length; i++)
                temp.Append(Convert.ToChar(rnd.Next(49, 58)));

            return temp.ToString();
        }

        /// <summary>
        /// ساخت یک عبارت رندوم
        /// </summary>
        public string CreateRandomName(int length)
        {
            var rnd = new Random((int) DateTime.Now.Ticks);
            var temp = new StringBuilder();
            var flag = 1;

            for (var i = 0; i < length; i++)
            {
                flag = rnd.Next(0, 15);

                if (flag < 5)
                    temp.Append(Convert.ToChar(rnd.Next(97, 123))); // lower
                else if (flag >= 5 && flag < 10)
                    temp.Append(Convert.ToChar(rnd.Next(49, 58))); // numeric
                else
                    temp.Append(Convert.ToChar(rnd.Next(65, 91))); // biger
            }

            return temp.ToString();
        }
}
```

همانطور که گفتم پیاده سازی متد های DecryptTpl و EncryptTpl کلاس CustomHashing رو به خود دوستان واگذار می‌کنم تا با هر الگوریتمی که دوست دارن این کار رو انجام بدن. (^)

امیدوارم کد بنده به دوستان کمک کنه.

موفق باشید

عنوان: **توزیع یک اپلیکیشن ASP.NET MVC 5 روی Windows Azure**
نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۰/۱۹ ۱۰:۵۰
آدرس: www.dotnettips.info
گروه‌ها: Tips, OpenID, ASP.NET MVC 5, ASP.NET Identity, Windows Azure, OAuth, MVC, Security

این مقاله به شما نشان می‌دهد چگونه یک اپلیکیشن وب ASP.NET MVC 5 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.
می توانید 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 تجربه ای 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 یاد می‌گیرید:
چطور یک اپلیکیشن وب ASP.NET MVC 5 بسازید و آن را روی یک وب سایت Windows Azure منتشر کنید.
چگونه از OpenID ، OAuth 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 بوده و از Entity Framework برای دسترسی داده استفاده می‌کند. تصویر زیر صفحه ورود نهایی اپلیکیشن را نشان می‌دهد.

http://localhost:12564/Accou
Log in - 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.
Facebook
Google
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
توزیع اپلیکیشن روی Windows Azure
افزودن یک دیتابیس به اپلیکیشن
افزودن یک OAuth Provider
استفاده از Membership API
توزیع 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 سایت‌های شما روی ماشین‌های مجازی (virtual machines) اجرا می‌شوند که با مشتریان دیگر Windows Azure به اشتراک گذاشته شده اند. یک محیط میزبانی اشتراکی گزینه ای کم هزینه برای شروع کار با رایانش‌های ابری است. اگر در آینده ترافیک وب سایت شما رشد چشم گیری 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 بر اساس نیازهای خود پیکربندی کنید.

Windows Azure SQL Database یک سرویس دیتابیس رابطه ای (relational) و مبتنی بر Cloud است که بر اساس تکنولوژی‌های SQL Server ساخته شده. ابزار و اپلیکیشن هایی 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 **Web Site** و سپس **Custom Create** 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 **Database** گزینه **Create a free 20 MB SQL 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 *DefaultConnection* 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 **Server** گزینه **New SQL Database Server** را انتخاب کنید. اگر قبلا دیتابیس 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 **Login Name** و **Password** مدیر (administrator) وارد کنید. اگر گزینه **New SQL Database server** را انتخاب کرده اید، چنین کاربری وجود ندارد و در واقع اطلاعات یک حساب کاربری جدید را وارد می‌کنید تا بعدا 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.

پرتال 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 (معمولا کمتر از یک دقیقه) این ستون نشان می‌دهد که سایت شما با موفقیت ایجاد شده. در منوی پیمایش سمت چپ، تعداد سایت هایی 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 ASP.NET MVC 5 بسازید

شما یک وب سایت Windows Azure ساختید، اما هنوز هیچ محتوایی 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 حتما 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. دکمه‌های 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 *Home* را با *Cm* جایگزین کنید تا از *CmController* 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="colla
                    <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
            </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 فعلا آماده است و می‌توانید آن را روی Windows Azure توزیع کنید. بعدا 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 **Publish Web** 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*.) که اطلاعات اکانت Windows Azure شما را دارد.

نکته امنیتی: این فایل تنظیمات را بیرون از پوشه‌های سورس کد خود دانلود کنید، مثلا 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 کاملا 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.

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. داده هایی که می‌خواهیم برای هر رکورد ذخیره کنیم تعریف شده اند، بعلاوه یک فیلد Primary Key که دیتابیس به آن نیاز دارد.

یک کنترلر و نما برای داده‌ها اضافه کنید

ابتدا پروژه را Build کنید (Ctrl + Shift+ B). این کار را باید پیش از استفاده از مکانیزم Scaffolding انجام دهید.

یک کنترلر جدید به پوشه Controllers اضافه کنید.

در دیالوگ باکس **Add Scaffold** گزینه **MVC 5 Controller with views, using EF** 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 **ApplicationDbContext** هم برای اطلاعات سیستم عضویت و هم برای داده‌های Contacts استفاده خواهد شد.

روی Add کلیک کنید. ویژوال استودیو بصورت خودکار با استفاده از Scaffolding متدها و Viewهای لازم برای عملیات CRUD را فراهم می‌کند، که همگی از مدل Contact استفاده می‌کنند.

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 **Library Package Manager** و سپس **Package Manager Console** 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 می‌کند. با استفاده از این کلاس می‌توانید داده‌های اولیه دیتابیس را وارد کنید و مهاجرت‌ها را نیز پیکربندی کنید.

در پنجره **Package Manager Console** فرمان زیر را وارد کنید.

```
add-migration Initial
```

فرمان **add-migration initial** فایلی با نام **<data_stamp> initial** ساخته و آن را در پوشه *Migrations* ذخیره می‌کند. در این مرحله دیتابیس شما ایجاد می‌شود. در این فرمان، مقدار **initial** اختیاری است و صرفا برای نامگذاری فایل مهاجرت استفاده شده. فایل‌های جدید را می‌توانید در **Solution Explorer** مشاهده کنید.

در کلاس **Initial** متد **Up** جدول Contacts را می‌سازد. و متد **Down** (هنگامی که می‌خواهید به وضعیت قبلی بازگردید) آن را drop می‌کند.

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 متد دیتابیس را Seed می‌کند، یعنی داده‌های پیش فرض 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 Fi
 model. This snapshot is used to calculate the changes to your model when you scaffol
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 لینک هایی 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.

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 internet از OAuth و OpenID استفاده می‌کند تا فیسبوک، توییتر، گوگل و حساب‌های کاربری مایکروسافت را بعنوان تامین کنندگان 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 کنندگان باید دنبال کنید، بسیار مشابه همین مراحلی است که در این مقاله دنبال خواهید کرد. برای اطلاعات بیشتر درباره نحوه استفاده از فیسبوک بعنوان یک تامین کننده احراز هویت به Create an ASP.NET MVC 5 App with Facebook and Google OAuth2 and OpenID Sign-on مراجعه کنید.

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 رکورد ها).

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 بدهید (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. نهایتا روی **Register** کلیک کنید.

Register - Contact Manag
localhost:62951/Account/ExternalLoginCall
CM Demo
Register.
Associate your Google account.
Association Form
You've successfully authenticated with Google. Please enter a user name for this site below and click the Register button to finish logging in.
User name
RickAnderson
Register
© 2013 - Contact Manager

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 مثلا *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، بجای 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;
```

متد **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 متد **Seed** این متد 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 *canEdit* و کاربری با نام *user1* می سازد. سپس این کاربر به نقش مذکور اضافه می‌شود.

کدی موقتی برای تخصیص نقش *canEdit* به کاربران جدید Social Provider ها

در این قسمت شما متد **ExternalLoginConfirmation** در کنترلر Account را ویرایش خواهید کرد. یا این تغییرات، کاربران جدیدی که توسط OAuth یا OpenID 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 مایکروسافت امیدوار است ابزاری مانند WSAT برای مدیریت کاربران و نقش‌ها در آینده عرضه کند. بعدا 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 متد **ExternalLoginConfirmation** را پیدا کنید.

درست قبل از فراخوانی **SignInAsync** متد **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(ExternalLoginConfirmati
{
    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 **Cm** و صفحات About و Contact نیز دسترسی خواهند داشت. همچنین دسترسی به متدهایی که داده‌ها را تغییر می‌دهند تنها برای کاربرانی وجود دارد که در نقش *canEdit* هستند.

خاصیت Authorize و RequireHttps 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 تا تمام دسترسی‌ها به اپلیکیشن وب 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. قسمتی از کد کامل شده در زیر آمده است.

```
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 چپ گزینه **Web** را انتخاب کنید. حالا مقدار **Project Url** را به آدرسی که کپی کرده اید تغییر دهید. نهایتا تغییرات را ذخیره کنید و پنجره را ببندید.

حال پروژه را اجرا کنید. مرورگر شما باید یک پیام خطای اعتبارسنجی به شما بدهد. دلیلش این است که اپلیکیشن شما از یک

Valid Certificate 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 قبلی 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 قبلا ساختید وارد سایت شوید. حال به صفحه ویرایشی که قبلا درخواست کرده بودید هدایت می‌شوید.

اگر نتوانستید با این کاربر به سایت وارد شوید، کلمه عبور را از سورس کد کپی کنید و مجددا 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 **Remote connection string** را برای **ApplicationDbContext** و دیتابیس ContactDB انتخاب کنید.

اگر ویژوال استودیو را پس از ساخت Publish 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.

Publish Web
cm22 *
Configuration:
Release
File Publish Options
Databases
ApplicationDbContext (DefaultConnection)
Remote connection string
ContactDB
Execute Code First Migrations (runs on application start)
< Prev
Next >
Publish
Close

زیر قسمت **ContactManagerContext** گزینه **Execute Code First Migrations** 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 **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
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
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.

فراخوانی 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. بعدا در ادامه مقاله، دسترسی حساب‌های کاربری محلی (local) را حذف می‌کنیم.

مطمئن شوید که به صفحات About و Contact دسترسی دارید.

لینک **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 قبلا 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.

دیتابیس SQL Azure را بررسی کنید

در **Server Explorer** دیتابیس **ContactDB** را پیدا کنید. روی آن کلیک راست کرده و **Open in SQL Server Object Explorer** را

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. دقت داشته باشید که در صورت اضافه کردن Port Rangeها 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
SQL Server
(localdb)\Projects (SQL Server 11.0.3000.0 -
ox6jx6raw7.database.windows.net (SQL Serv
Databases
ContactDB
Tables
System Tables
dbo.__MigrationHistory
dbo.AspNetRoles
dbo.AspNetUserClaims
dbo.AspNetUserLogins
dbo.AspNetUserRoles
dbo.AspNetUsers
Data Comparison...
Script As
View Code
View Designer
View Permissions
View Data
Delete
Del
Rename
Refresh
Properties

dbo.AspNetUsers [Data]

Max Rows: 1000

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	ACDwfL01Klf...	2977f...	ApplicationUser
NULL	NULL	NULL	NULL	NULL

حالا روی **AspNetUserRoles** کلیک راست کنید و **View Data** را انتخاب کنید.

dbo.AspNetUserRoles [Data] | dbo.AspNetUsers [Data]

Max Rows: 1000

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 دارند.

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.

هنگامی که با پیغام "The current IP address xxx.xxx.xxx.xxx is not included in existing firewall rules. Do you want? to update the firewall rules" مواجه شدید **Yes** را کلیک کنید. افزودن یک آدرس IP بدین روش معمولا کافی نیست و در فایروال‌های سازمانی و بزرگ باید Range بیشتری را تعریف کنید.

مرحله بعد اضافه کردن محدوده آدرس‌های مجاز است.

مجددا در پرتال مدیریتی Windows Azure روی **SQL Databases** کلیک کنید. سروری که دیتابیس شما را میزبانی می‌کند انتخاب کنید.

در بالای صفحه لینک **Configure** را کلیک کنید. حالا نام rule جدید، آدرس شروع و پایان را وارد کنید.

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 **SQL Databases** را انتخاب کنید. دیتابیس **ContactDB** را انتخاب کرده و روی **View SQL Database connection strings** کلیک کنید. در صفحه **Connection Strings** مقادیر **Server** و **User ID** را کپی کنید. حالا مقادیر را 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 **SQL Server Object Explorer** روی جدول **AspNetUserRoles** کلیک راست کنید و **View Data** را انتخاب کنید.

dbo.AspNetUserRoles [Data] | dbo.AspNetUsers [Data]

Max Rows: 1000

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.

dbo.AspNetUserRoles [Data] | dbo.AspNetUsers [Data]

Max Rows: 1000

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 بازنشانی کلمه‌های عبور (password reset) پشتیبانی نمی‌کند. همچنین اطمینان حاصل نمی‌شود که کاربران سایت انسان هستند (مثلا 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 پوشه *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

OpenID Sign-on

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 پیشرفته EF ارائه می‌کند.

نظرات خوانندگان

نویسنده: مهمان
تاریخ: ۱۳۹۲/۱۰/۱۹ ۱۴:۴

دوست عزیز

با صبر و حوصله و دقت فراوان یک مقاله خوب را منتشر کردید. ممنون(رای من 5)

عنوان: **MVC Scaffolding #1**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۰۲ ۱۳:۱۵
آدرس: www.dotnettips.info
گروه‌ها: Entity framework, MVC, MVC Scaffolding

پیشنیازها

کل سری ASP.NET MVC

به همراه کل سری EF Code First

MVC Scaffolding چیست؟

MVC Scaffolding 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 اکشن متدهای 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 DbContext مرتبط، گزینه new data context را انتخاب کرده و نام پیش فرض آن‌را پذیرفته‌ایم. زمانیکه بر روی دکمه Add کلیک کنیم، اتفاقات ذیل رخ خواهند 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 معرض دید EF Code first قرار دهد. (همانطور که عنوان شد یکی از پیشنیازهای بحث Scaffolding آشنایی با EF Code first است)

ج) در پوشه Views\Tasks، پنج 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 باشید (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:

- DbContext در سطح یک کنترلر وهله سازی شده و الگوی Context Per Request در اینجا بکارگرفته نشده است. واقعیت یک برنامه ASP.NET MVC کامل، داشتن چندین Partial View تغذیه شونده از کنترلرهای مختلف در یک صفحه واحد است. اگر قرار باشد به ازای هر کدام یکبار DbContext وهله 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 متدها استفاده نشده است.
- از ViewModel تعریف شده‌ای به نام Task هم به عنوان Domain model و هم ViewModel استفاده شده است. یک کلاس متناظر با جداول بانک اطلاعاتی می‌تواند شامل فیلدهای بیشتری باشد و نباید آن‌را مستقیما در معرض دید یک View قرار داد (خصوصا از لحاظ مسایل امنیتی).

مزیت‌ها:
قسمت عمده‌ای از کارهای «اولیه» تهیه یک کنترلر و همچنین Viewهای مرتبط به صورت خودکار انجام شده‌اند. کارهای اولیه‌ای که با هر روش و الگوی شناخته شده‌ای قصد پیاده سازی آن‌ها را داشته باشید، وقت زیادی را به خود اختصاص داده و نهایتا آنچنان تفاوت عمده‌ای هم با کدهای تولیدی در اینجا نخواهند داشت. حداکثر فرم‌های آن‌را بخواهید با jQuery Ajax پیاده سازی کنید یا کنترل‌های پیش فرض را با افزونه‌های jQuery غنی سازی نمائید. اما شروع کار و کدهای اولیه چیزی بیشتر از این نیست.

نصب بسته اصلی MVC Scaffolding توسط NuGet

بسته اصلی MVC Scaffolding را با استفاده از دستور خط فرمان Powershell ذیل، از طریق منوی Tools، گزینه Library package manager و انتخاب Package manager console می‌توان 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 می‌کند:

```
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 شده‌اند در قسمت‌های 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 می‌توان پیاده سازی کرد:

```
Scaffold Controller Task
```

نوشتن این دستور نیز ساده است. حروف sca را تایپ کرده و دکمه tab را فشار دهید. منویی ظاهر خواهد شد که امکان انتخاب دستور Scaffold را می‌دهد. یا برای نوشتن Controller نیز به همین نحو می‌توان عمل کرد.
نکته و مزیت مهم دیگری که در اینجا در دسترس می‌باشد، سوئیچ‌های 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. منویی ظاهر خواهد شد که بیانگر سوئیچ‌های قابل استفاده است.

```
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

PM> Scaffold Controller -
```

```
-OutVariable
-OutBuffer
-ControllerName
-ModelType
-CodeLanguage
-DbContextType
-Area
-ViewScaffolder
-Layout
```

```
plication1.
tion1.
ation1.
```

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 مبنی بر اینکه Viewهای Edit و Create بسیار شبیه به هم هستند، ایجاد شده است.

بهبود مقدماتی کیفیت کد تولیدی MVC Scaffolding

در همان کنسول پاروشل NuGet، کلید up arrow را فشار دهید تا مجددا 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 مبنی 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 بی‌کیفت ابتدایی همراه با وهله 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؛ دقیقا 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 IQueryableها 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

نویسنده: حسین
تاریخ: ۱۳۹۱/۱۱/۰۲ ۱۳:۴۹

فوق العاده بود. اصلاً نمیدونستم که Scaffolding یه همچین قابلیت هایی هم داره.
ممنون.

نویسنده: سعید یزدانی
تاریخ: ۱۳۹۱/۱۱/۰۲ ۱۴:۱۸

با تشکر
در کل از این روش در تولید پروژه‌های واقعی استفاده میشود ؟

نویسنده: سعید
تاریخ: ۱۳۹۱/۱۱/۰۲ ۱۶:۲۰

حداقل 4 بار در این متن کلمه «اولیه» بکار رفته؛ به همراه گیومه دورش. حتما دلیلی داشته ...

نویسنده: پژمان پارسائی
تاریخ: ۱۳۹۱/۱۱/۰۳ ۶:۰

دست مریزاد. خیلی مفید بود. ممنون

نویسنده: 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'.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۰۴ ۱۷:۲۷

از EF 6 استفاده کردید؟ بله . فقط برای MVC 5 ابزار Scaffolding را جهت کار با EF 6 به روز کرده‌اند .

عنوان: **MVC Scaffolding #2**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۰۳ ۸:۱۰
آدرس: www.dotnettips.info
برچسب‌ها: Entity framework, MVC, MVC Scaffolding

از آنجائیکه 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 هرکدام در دستور نهایی سبب true شدن مقدار آن‌ها می‌گردد:
Repository: برای تولید کدها بر اساس الگوی مخزن
Force: برای بازنویسی فایل‌های موجود.
ReferenceScriptLibraries: ارجاعاتی را به اسکریپت‌های موجود در پوشه Scripts، اضافه می‌کند.
NoChildItems: در این حالت فقط کلاس کنترلر تولید می‌شود و از سایر ملحقات مانند تولید Viewها، DbContext 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 هربار 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 آن‌را 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 خواهند شد. اگر در اینجا برای نمونه Controller انتخاب شود، مجددا با ورود یک فاصله و خط تیره و سپس فشردن دکمه Tab، لیست پارامترهای مجاز و همراه با سوئیچ کنترلر ظاهر می‌گردند.

MVC Scaffolding و مدیریت روابط بین کلاس‌ها

مثال قسمت قبلی بسیار ساده و شامل یک کلاس بود. اگر آن‌را کمی پیچیده‌تر کرده و برای مثال روابط one-to-many و 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 Task، Tag 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؛ خصوصا 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 CRUD دو متد را ایجاد کنیم. یکی برای نمایش و دیگری برای ثبت. بعد بر روی هر متد کلیک راست کرده و Viewهای متناظری را ایجاد کنیم. سپس مجددا یک سری پیاده سازی «مقدماتی» تکراری را به ازای هر متد جهت ثبت یا ذخیره اطلاعات تدارک ببینیم. 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 می‌گردند.

چند نکته:

- با توجه به اینکه مدل‌ها تغییر کرده‌اند، نیاز است بانک اطلاعاتی متناظر نیز به روز گردد. مطالب مرتبط با آن‌را در مباحث Migrations 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 به خوبی با View متناظری دارای یک drop down list تولید خواهد شد. در اینجا لیست تولیدی به صورت خودکار با مقادیر خاصیت Name کلاس Status پر می‌شود. اگر این نام دقیقا 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 MVC Scaffolding

MVC Scaffolding 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 متدی به نام ArchiveTask با استفاده از کلاس ViewModel ایی به نام Task اجرا می‌شود.
حاصل آن ایجاد اکشن متد یاد شده به همراه کلاس TasksControllerTest است؛ البته اگر حین ایجاد پروژه جدید در ابتدای کار، گزینه ایجاد پروژه آزمون‌های واحد را نیز انتخاب کرده باشید. نام پروژه پیش فرضی که جستجوی می‌شود YourMvcProjectName.Test/Tests است.
نکته مهم آن، عدم حذف یا بازنویسی کامل کنترلر یاد شده است. کاری هم که در تولید متد آزمون واحد متناظر انجام می‌شود، تولید بدنه متد 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 متد آزمون واحد کنترلر Tasks و اکشن متد Delete آن، تولید می‌شود.

کار مقدماتی با MVC Scaffolding و امکانات مهیای در آن همینجا به پایان می‌رسد. در قسمت‌های بعد به سفارشی سازی این مجموعه خواهیم پرداخت.

نظرات خوانندگان

نویسنده: سهیلا صالح زاده
تاریخ: ۱۳۹۲/۰۶/۲۳ ۱۶:۰

در بخش EF Code First #11 عنوان کردید که مایکروسافت در تعریف DbContext اعلام می‌کند که DbSetها 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 . خطا میدهد ولی اگر کلاس‌ها در یک پروژه تعریف شوند مشکلی ندارد.

نویسنده: سهیلا 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 ویا 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 دقیقا مانند نحوه مقدار دهی نوع مدل در صفحه دیالوگ استاندارد اضافه کردن یک View در VS.NET مقدار دهی می‌شود؛ یک fully qualified name است. 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؛ مثلا کد زیر در پروژه DataLayer به درستی کار می‌کنه اما در پروژه اصلی با وجود Add شدن Refrence پروژه DataLayer کار نمی‌کنه و خطا میده.

مجبور 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 (... Task ... - *ModelType*)

- خطاهایی رو هم که دریافت می‌کنید، اینجا به نویسنده اصلی گزارش بدید (به صورت کامل البته؛ نه اینکه صرفا عنوان کنید کار نمی‌کند).

عنوان: **MVC Scaffolding #3**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۰۴ ۸:۵
آدرس: www.dotnettips.info
گروه‌ها: Entity framework, MVC, MVC Scaffolding

شاید کیفیت کدهای تولیدی یا کدهای View حاصل از MVC Scaffolding مورد تائید 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 پوشه‌های MvcScaffolding، T4Scaffolding و T4Scaffolding.Core ساختار اصلی این بسته را تشکیل می‌دهند. برای نمونه اگر پوشه T4Scaffolding\tools را باز کنیم، شاهد تعدادی فایل ps1 خواهیم بود که همان فایل‌های پاورشل هستند. مطابق طراحی NuGet، همواره فایلی با نام init.ps1 در ابتدا اجرا خواهد شد. همچنین در اینجا پوشه‌های T4Scaffolding\tools\EFRepository و T4Scaffolding\tools\EFDbContext نیز قرار دارند که حاوی قالب‌های اولیه کدهای مرتبط با الگوی مخزن و DbContext تولیدی می‌باشند.
در پوشه MvcScaffolding\tools، ساختار قالب‌های پیش فرض تولید Viewها و کنترلرهای تولیدی قرار دارند. در اینجا به ازای هر مورد، دو نگارش vb و cs قابل مشاهده است.

سفارشی سازی قالب‌های پیش فرض Viewهای MVC Scaffolding

برای سفارشی سازی قالب‌های پیش فرض از دستور کلی زیر استفاده می‌شود:

```
Scaffold CustomTemplate Name Template
```

مانند دستور زیر:

```
Scaffold CustomTemplate View Index
```

در اینجا View نام یک Scaffolder است و Index نام قالبی در آن.
اگر دستور فوق را اجرا کنیم، فایل جدیدی به نام CodeTemplates\Scaffolders\MvcScaffolding.RazorView\Index.cs.t4 به پروژه جاری اضافه می‌شود. از این پس کلیه فرامین اجرایی، از نسخه محلی فوق بجای نمونه‌های پیش فرض استفاده خواهند کرد.
در ادامه قصد داریم اندکی این قالب پیش فرض را جهت اعمال ویژگی DisplayName به هدر جدول تولیدی نمایش اطلاعات Tasks تغییر دهیم. در کلاس Task، خاصیت زمان موعود با ویژگی DisplayName مزین شده است. این نام نمایشی حین تولید فرم‌های ثبت و ویرایش اطلاعات بکار گرفته می‌شود، اما در زمان تولید جدول اطلاعات ثبت شده، به هدر جدول اعمال نمی‌گردد.

```
[DisplayName("Due Date")]
public DateTime? DueDate { set; get; }
```

برای تغییر و بهبود این مساله، فایل Index.cs.t4 را که پیشتر به پروژه 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، متد 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 متد GetDisplayName، متد GetEligibleProperties را یافته و به نحو زیر تغییر 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 = GetModelProperties(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 بجای 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 از فایل ControllerWithRepository.cs.t4 موجود در پوشه packages\MvcScaffolding\tools\Controller گرفته شده است.

به این ترتیب فایل جدید CodeTemplates\Scaffolders\MvcScaffolding.Controller\ControllerWithRepository.cs.t4 به پروژه جاری اضافه 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 ForceMode با مقدار ControllerOnly استفاده شده است.

یا 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 Scaffolderهای مهیا با دستور Get-Scaffolder قابل مشاهده است.

نظرات خوانندگان

نویسنده: محسن عباس آباد عربی
تاریخ: ۱۳۹۱/۱۱/۰۴ ۹:۵۲

مرسی از مطلب مفیدتون
یا علی.

نویسنده: حسینی
تاریخ: ۱۳۹۱/۱۱/۲۰ ۱:۵

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: ۱۳۹۱/۱۱/۲۱ ۱۹:۵۴

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: ۱۳۹۲/۰۳/۰۸ ۱:۲۲

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: ۱۳۹۲/۰۹/۱۵ ۱۴:۱۹

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: ۱۳۹۳/۰۱/۲۲ ۲۲:۱۷

من template رو تغییر دادم و DisplayNameها جایگزین PropertyNameها 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: ۱۳۹۳/۰۱/۲۲ ۲۲:۲۶

encoding را به این نحو باید تنظیم کرد:

```
<#@ output extension=".cs" encoding="utf-8" #>
```

« نحوه استفاده از Text templateها در دات نت - قسمت سوم »

عنوان: **معماری لایه بندی نرم افزار #1**
نویسنده: میثم خوشبخت
تاریخ: ۱۳۹۱/۱۲/۲۸ ۱۸:۵
آدرس: www.dotnettips.info
گروه‌ها: Design patterns, SoC, Separation of Concerns, ASP.Net, Domain Driven Design, DDD, SOLID Principals, C#, MVC, WPF, N-Layer Architecture

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 مطمئنا 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 داده ها، منطق تجاری و سایر نیازهای برنامه کاربردی، کد نویسی خواهند 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 عمده‌ی 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 جداسازی کدها در پروژه‌های متفاوت انجام شود. در شکل زیر نمونه ای 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 شیوه‌های پیاده سازی جدیدی مواجه شوید که این نوع پیاده سازی برای شما قابل درک نباشد. اگر کمی صبر پیشه نمایید و این مجموعه‌ی 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 بخشهای مختلف برنامه بین اعضای تیم و سایر مزایای فوق العاده آن می‌باشد.

1- Visual Studio را باز کنید و یک Solution خالی با نام SoCPatterns.Layered ایجاد نمایید.

- جهت ایجاد Solution خالی، پس از انتخاب New Project ، از سمت چپ گزینه Other Project Types و سپس Visual Studio Solutions را انتخاب نمایید. از سمت راست گزینه Blank Solution را انتخاب کنید.

2- 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.Model و SoCPatterns.Layered.Repository ارجاع دهید.

-7 بر روی پروژه SoCPatterns.Layered.Presentation کلیک راست نمایید و با انتخاب گزینه Add Reference به پروژه‌های SoCPatterns.Layered.Model و SoCPatterns.Layered.Service ارجاع دهید.

-8 بر روی پروژه‌ی UI خود به عنوان مثال SoCPatterns.Layered.WebUI کلیک راست نمایید و با انتخاب گزینه Add Reference به پروژه‌های SoCPatterns.Layered.Service ، SoCPatterns.Layered.Repository ، SoCPatterns.Layered.Model و SoCPatterns.Layered.Presentation ارجاع دهید.

-9 بر روی پروژه‌ی 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.

1. UI
§ SoCPatterns.Layered.WebUI

2. Presentation Layer
§ SoCPatterns.Layered.Presentation

3. Service Layer
§ SoCPatterns.Layered.Service

4. Domain Layer

§ SoCPatterns.Layered.Model

5. Data Layer

§ 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: ۱۳۹۱/۱۲/۲۸ ۲۰:۲

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: ۱۳۹۱/۱۲/۲۸ ۲۲:۵

آیا صرفا تعریف چند ماژول 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 خوشبخت
تاریخ: ۱۳۹۱/۱۲/۲۹ ۰:۵۳

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 شبهات برطرف بشه. در مورد مرزبندی لایه‌ها هم صحبت می‌کنم و مفهوم هر کدام را دقیقا توضیح میدم.

نویسنده: میثم خوشبخت
تاریخ: ۱۳۹۱/۱۲/۲۹ ۰:۵۹

اگر مقاله فوق رو با دقت بخونید متوجه میشید که MVC و MVVM در لایه UI پیاده سازی میشن. البته در MVC لایه Model رو به Domain Model و Repository در برخی مواقع لایه Controller رو در لایه Presentation قرار میدن. در MVVM نیز لایه Model در Domain Model و Repository و لایه View Model نیز در لایه Presentation قرار میگیره. همچنین View Modelها نیز در لایه Service قرار میگیرن.
در مورد ماژول بندی هم اگر در مقاله خونده باشید میتونید لایه‌ها رو از طریق پوشه ها، فضای نام و یا پروژه‌ها از هم جدا کنید

نویسنده: حسن
تاریخ: ۱۳۹۱/۱۲/۲۹ ۱۰:۱۴

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 صرفا تعریف چند ماژول جدید می‌تواند ضمانتی باشد بر رفع مشکل code behind یا اینکه با این ماژول‌ها هم نهایتا همان مشکل قبل پابرجا است یا می‌تواند پابرجا باشد.

ضمن اینکه تعریف شما از لایه دقیقا چی هست؟ به نظر فقط تعریف یک اسمبلی در اینجا لایه نام گرفته.

نویسنده: آرمان فرقانی
تاریخ: ۱۳۹۱/۱۲/۲۹ ۱۱:۵۸

صحبت شما کاملاً صحیح است و صرفاً با ماژولار 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 کد 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. ماژولار کردنه ممکنه مشکل Smart UI 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 مواقع میتونه مفید باشه.
ممنون دوست عزیز

نویسنده: مسعود مشهدی
تاریخ: ۱۳۹۲/۰۱/۰۴ ۱۸:۳۳

با سلام
بابت مطالبتون سپاسگذارم
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
```

Application.Common

با تشکر

عنوان: معماری لایه بندی نرم افزار #2
نویسنده: میثم خوشبخت
تاریخ: ۱۳۹۱/۱۲/۳۰ ۱:۴۵
آدرس: www.dotnettips.info
برچسب‌ها: ASP.Net, C#, Design patterns, MVC, WPF, SoC, Separation of Concerns, Domain Driven Design, DDD, SOLID Principals, N-Layer Architecture

Business Layer یا Domain Model

پیاده سازی را از منطق تجاری یا Business Logic آغاز می‌کنیم. در روش کد نویسی Smart UI ، منطق تجاری در Code Behind قرار می‌گرفت اما در روش لایه بندی، منطق تجاری و روابط بین داده‌ها در Domain Model طراحی و پیاده سازی می‌شوند. در مطالب بعدی راجع به Domain Model و الگوهای پیاده سازی آن بیشتر صحبت خواهم کرد اما بصورت خلاصه این لایه یک مدل مفهومی از سیستم می‌باشد که شامل تمامی موجودیت‌ها و روابط بین آنهاست.

الگوی Domain Model جهت سازماندهی پیچیدگی‌های موجود در منطق تجاری و روابط بین موجودیت‌ها طراحی شده است.

شکل زیر مدلی را نشان می‌دهد که می‌خواهیم آن را پیاده سازی نماییم. کلاس Product موجودیتی برای ارائه محصولات یک فروشگاه می‌باشد. کلاس Price جهت تشخیص قیمت محصول، میزان سود و تخفیف محصول و همچنین استراتژی‌های تخفیف با توجه به منطق تجاری سیستم می‌باشد. در این استراتژی همکاران تجاری از مشتریان عادی تفکیک شده اند.

Domain Model را در پروژه SoCPatterns.Layered.Model پیاده سازی می‌کنیم. 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 می‌باشد که در مطالب بعدی در مورد این الگو بیشتر صحبت خواهم 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 معمولا 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 الگوها، 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 Setter Injection در متد 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 شدند 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 اتباط 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 IProductRepository را با کد زیر به پروژه SoCPatterns.Layered.Model اضافه کنید:

```
public interface IProductRepository
{
    IList<Product> FindAll();
}
```

الگوی Repository به عنوان یک مجموعه‌ی در حافظه (In-Memory Collection) یا انباره ای از موجودیت‌های تجاری عمل می‌کند که نسبت به زیر بنای ساختاری منبع داده ای کاملا مستقل می‌باشد.

کلاس سرویس باید بتواند استراتژی تخفیف را بر روی مجموعه ای 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 کاملا نامطلع می‌باشد و می‌تواند برنامه نویس را به سمت Abstraction و Dependency Inversion به جای پیاده سازی واقعی سوق دهد.

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 Domain Model به هیچ منبع داده ای خاصی وابسته نیست و به جای پیاده سازی کدهای منبع داده ای، از Interface ها به منظور برقراری ارتباط با پایگاه داده استفاده شده است. پیاده سازی کدهای منبع داده ای 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 کد نویسی‌های منبع داده ای نشود و بتواند به صورت مستقل و فارغ از بخشهای مختلف برنامه تست شود. لایه بعدی که می‌خواهیم کد نویسی آن را آغاز کنیم، لایه‌ی Service می‌باشد.

در کد نویسی‌های فوق از الگوهای طراحی (Design Patterns) متعددی استفاده شده است که به صورت مختصر در مورد آنها صحبت کردم. اصلا جای نگرانی نیست، چون در مباحث بعدی به صورت مفصل در مورد آنها صحبت خواهم 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
ممنون از شما این بخش هم کامل و زیبا بود
ولی کمی فشرده بود
لطفا اگر ممکن هست در مورد معماری‌ها و الگوها و بهترین‌های آنها کمی توضیح دهید یا منبعی معرفی کنید تا این الگوها و معماری برای ما بیشتر مفهوم بشه
من در این زمینه تازه کارم و از شما میخواهم که من رو راهنمایی کنید که چه مقدماتی در این زمینه‌ها نیاز دارم
باز هم ممنون.

نویسنده: علي
تاریخ: ۱۳۹۱/۱۲/۳۰ ۹:۱۲

در همین سایت مباحث الگوهای طراحی و Refactoring مفید هستند.

و یا الگوهای طراحی Agile رو هم در اینجا می‌تونید پیگیری کنید.

نویسنده: میثم خوشبخت
تاریخ: ۱۳۹۱/۱۲/۳۰ ۱۱:۳۸

فشردگی این مباحث بخاطر این بود که میخواستم فعلا یک نمونه پروژه رو آموزش بودم تا یک شمای کلی از کاری که می‌خواهیم انجام بدیم رو ببینید. در مباحث بعدی این مباحث رو بازتر می‌کنم. خود من برای مطالعه و جمع بندی این مباحث منابع زیادی رو مطالعه کردم. واقعا برای بعضی مباحث 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 الحاقی 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 را فراخوانی کند و به این شکل کلاس‌های بیرونی رو از تغییرات درونی کلاس Product مستقل کنیم ؟

نویسنده: میثم خوشبخت
تاریخ: ۱۳۹۱/۱۲/۳۰ ۱۸:۱

دوست عزیزم. متد Apply یک Extension Method برای IList<Product> است. اگر این متد تعریف نمی‌شد شما باید در کلاس سرویس حلقه 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، واقعا ممنونم آقای خوشبخت، لطفا 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 واقعا جای تشکر داره.

نویسنده: f.tahan36
تاریخ: ۱۳۹۲/۰۲/۲۹ ۱۷:۱۰

با سلام

تفاوت factory با design factory در 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 تمرین کلاسیت رو آوردی 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 داده ای قوی (Strongly Typed) به نام View Model ، برای لایه‌ی Presentation فراهم می‌کند. کلاس View Model یک Strongly Typed 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 خواهم کرد.

الگوی Facade یک Interface ساده را به منظور کنترل دسترسی به مجموعه ای از Interface ها و زیر سیستم‌های پیچیده ارائه می‌کند. در مباحث بعدی در مورد آن بیشتر صحبت خواهم 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 Request 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 ، به دو متد نیاز داریم، یکی برای تبدیل یک Product و دیگری برای تبدیل لیستی از Product . شما می‌توانید این دو متد 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 می‌نماییم. 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 خطاها را دریافت نموده و پس از مدیریت خطا، پیغامی مناسب را به کلاینت ارسال می‌کند. همچنین این لایه محل مناسبی برای Log کردن خطاها می‌باشد. در اینجا کد نویسی لایه سرویس به پایان رسید و در ادامه به کدنویسی Data Layer می‌پردازیم.

Data Layer

برای ذخیره سازی محصولات، یک بانک اطلاعاتی با نام Shop01 ایجاد کنید که شامل جدولی به نام Product با ساختار زیر باشد:

Product

Column Name	Data Type	Allow Nulls
ProductId	int	☐
ProductName	nvarchar(50)	☐
Rrp	smallmoney	☐
SellingPrice	smallmoney	☐
		☐

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.Layered.Repository 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 متد FindAll ، با استفاده از دستورات Linq to SQL ، لیست تمامی محصولات را برگرداندیم. کدنویسی لایه‌ی Data هم به پایان رسید و در ادامه به کدنویسی لایه‌ی Presentation و UI می‌پردازیم.

Presentation Layer

به منظور جداسازی منطق نمایش (Presentation) از رابط کاربری (User Interface) ، از الگوی Model View Presenter یا همان MVP استفاده می‌کنیم که در مباحث بعدی با جزئیات بیشتری در مورد آن صحبت خواهم کرد. یک Interface با نام IProductListView را با کد زیر به پروژه SoCPatterns.Layered.Presentation اضافه کنید:

```
using SoCPatterns.Layered.Service;

public interface IProductListView
{
    void Display(IList<ProductViewModel> Products);
    Model.CustomerType CustomerType { get; }
    string ErrorMessage { set; }
}
```

این Interface توسط Web Form های ASP.NET و یا Win Form ها باید پیاده سازی شوند. کار با Interface ها موجب می‌شود 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 IProductListView _productListView;
        private Service.ProductService _productService;
        public ProductListPresenter(IProductListView ProductListView,
            Service.ProductService ProductService)
        {
            _productService = ProductService;
            _productListView = ProductListView;
        }
        public void Display()
        {
            ProductListRequest productListRequest = new ProductListRequest();
            productListRequest.CustomerType = _productListView.CustomerType;
            ProductListResponse productResponse =
                _productService.GetAllProductsFor(productListRequest);
            if (productResponse.Success)
            {
                _productListView.Display(productResponse.Products);
            }
            else
            {
                _productListView.ErrorMessage = productResponse.Message;
            }
        }
    }
}
```

کلاس Presenter وظیفه‌ی واکشی داده ها، 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 نویسی Web Form ها باشید. 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

ممنون 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 تعویض شود؟ 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 می‌کنند. یعنی شاید بهتر باشه قسمت مخزن و همچنین سرویس یکی بشن.
- چرا لایه سرویس تعریف شده از یک یا چند اینترفیس مشتق نمی‌شود؟ اینطوری تهیه تست برای اون ساده‌تر میشه. همچنین پیاده سازی‌ها هم وابسته به یک کلاس خاص نمی‌شن چون از اینترفیس دارن استفاده می‌کنند.
- این اشیاء Request و Response هم در عمل به نظر نوعی ViewModel هستند. درسته؟ اگر اینطوره بهتر یک مفهوم کلی دنبال بشه تا سردرگمی‌ها رو کمتر کنه.

یک سری نکته جانبی هم هست که می‌تونه برای تکمیل بحث جالب باشه:
- مثلا الگوی Context per request بجای نوشتن new ShopDataContext بهتر استفاده بشه تا برنامه در طی یک درخواست در یک تراکنش و اتصال کار کنه.
- در مورد try/catch و استفاده از اون بحث زیاد هست. خیلی‌ها توصیه می‌کنن که یا اصلا استفاده نکنید یا استفاده از اون‌ها رو به بالاترین لایه برنامه موکول کنید تا این وسط کرش یک قسمت و بروز استثناء در اون، از ادامه انتشار صدمه به قسمت‌های بعدی جلوگیری کنه.

نویسنده: میثم خوشبخت
تاریخ: ۱۳۹۲/۰۱/۰۲ ۲۳:۳۵

محسن عزیز. از شما ممنونم که به نکته‌های ظریفی 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 مناسب‌تر خواهد بود.
در مورد ORM هم من با چند Application سروکار داشتم که در روال توسعه بخش‌های جدید رو بنا به دلایلی با ORM یا DB متفاوتی توسعه داده اند. غیر از این موضوع، حتی بخشهایی از مدل، سرویس و یا مخزن رو در پروژه‌های دیگری استفاده کرده اند. همچنین برخی از نکات مربوط به تفکیک لایه‌ها به منظور تست پذیری راحت‌تر رو هم در نظر بگیرید.
در مورد اشیا Request و Response هم باید خدمتتان عرض کنم که برای درخواست و پاسخ به درخواست استفاده می‌شوند که چون پروژه ای که مثال زدم کوچک بوده ممکنه کاملا درکش نکرده باشید. ما کلاسهای Request و Response متعددی در پروژه داریم که ممکنه خیلی از اونها فقط از یک View Model استفاده کنن ولی پارامترهای 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 میتونیم 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 لایه‌ی Data نشدم. در مورد اون اگه اجازه بدی بعدا صحبت میکنم.

نویسنده: ایلیا
تاریخ: ۱۳۹۲/۰۱/۰۳ ۰:۴۳

آقای خوشبخت خداقوت.

مرسی از مطالب خوبتون.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۱/۰۳ ۰:۴۸

لطفا برای اینکه نظرات حالت فنی‌تر 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 مثلا EF یا LINQ to SQL معرفی می‌کنند. فرضشون هم این است که این روش ما رو از تماس مستقیم با ORM برحذر 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 با ORMهای دیگر با روش‌های مختلف و بیشتر یک تصور واهی هست که جنبه عملی نداره. بیشتر تئوری هست بدون پایه تجربه دنیای واقعی. ضمن اینکه این روکش باعث میشه نتونید از خیلی از امکانات ORM مورد استفاده 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: افشین
تاریخ: ۱۳۹۲/۰۱/۰۶ ۱۱:۱۵

لطفا دمو یا سورس برنامه رو هم قرار بدید که یادگیری و آموزش سریعتر انجام بشه.

ممنون

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 پروژه شما ممنون میشم شرح بیشتری بدین که جایگاه پیاده سازی الگو واحد کار با توجه به مزایایی که دارد در کدام لایه است؟

نویسنده: رام
تاریخ: ۱۳۹۲/۰۱/۱۶ ۵:۲۹

محسن جان، چیزی که من از این الگو در مورد واکشی و نمایش داده‌ها برداشت میکنم اینه:

کلاس‌های لایه مخزن با دریافت دستور از لایه سرویس آبجکت مدل مربوطه را پر میکنند و به بالا (لایه سرویس) پاس میدند.

بعد

در لایه سرویس نمونه‌ی مدل مربوطه به ویـــــومدل متناظر باهاش تبدیل میشه و به لایه بالاتر فرستاده میشه

بنابراین
کار در **"لایه مخزن"** روی **"مدل ها"** انجام میگیره
و
کار در **"لایه سرویس"** روی **"ویـــــومدل 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

نویسنده: رام
تاریخ: ۱۳۹۲/۰۱/۱۶ ۵:۴۹

شاهین جان، من با حذف لایه مخزن مخالف هستم. زیرا:

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. حالا این اطلاعات ممکنه از پایگاه داده یا جاهای دیگه جمع آوری بشوند (و الزاما توسط EF قابل دسترسی و ارائه نباشند)

همچنین گاهی نیاز هست که بر مبنای چند متد که EF به ما میرسونه (مثلا چند SP) یک متد کلی‌تر را تعریف کنیم (چند فراخوانی را در یک متد مثلا متد X در لایه مخزن انجام دهیم) و در لایه بالاتر آن متد را صدا بزنیم (بجای نوشتن و تکرار پاپی همه کدهای نوشت شده در متد 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 گاهی نیاز میشه.

بنابراین:
من وجود لایه مخزن را ضروری میدونم.

(فراموش نکنیم که هدف از این آموزش تعریف یک الگوی معماری مناسب برای پروژه‌های بزرگ هست و الا بدون خیلی از اینها هم میشه برنامه ساخت. همونطور که اکثرا بدون این ساختارها و خیلی ساده‌تر میسازند)

نویسنده: محسن
تاریخ: ۱۳۹۲/۰۱/۱۶ ۹:۳

- بحث آقای شاهین و من در مورد مثال عینی بود که زده شد. در مورد کار با ORM که کدهاش دقیقا ارائه شده. این روش قابل نقد و رد است.

شما الان اومدی یک بحث انتزاعی کلی رو شروع کردید. بله. اگر ORM رو کنار بگذارید مثلا می‌رسید به ADO.NET (یک نمونه که خیلی‌ها در این سایت حداقل یکبار باهاش کار کردن). این افراد پیش از اینکه این مباحث مطرح باشن برای خودشون لایه DAL داشتند و تمام جزئیات ADO.NET رو کپسوله کرده بودن در اون. حالا با اومدن ORMها این لایه DAL کنار رفته چون خود ORM 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 مباحث ربطی به یک ORM تمام عیار که کپسوله کننده ADO.NET است ندارد.

- ترکیب چند SP در لایه مخزن انجام نمیشه. چیزی رو که عنوان کردید یعنی پیاده سازی منطق تجاری و این مورد باید در لایه سرویس باشه. اگر از ADO.NET استفاده میشه، می‌تونیم 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 متد انجام شده. بنابراین بهتر است وضعیت و سطح لایه‌ای رو که داریم باهاش کار می‌کنیم خوب بررسی و درک کنیم.

- می‌تونید 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 مثلا یک نمونه: ORMها لایه‌ای دارند به نام سطح اول کش که مثلا در EF اسمش هست Trackig API. این لایه فقط در حین کار با Context همون ORM کار می‌کنه. اگر دو مورد رو با هم مخلوط کنید، قابل استفاده نیست، ترکیب پذیر نیستند. از این دست باز هم هست مثلا 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: ۱۳۹۲/۰۱/۱۶ ۹:۱۵

سعی نکنید انتزاعی بحث کنید. چون در این حالت این حرف می‌تونه 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 می‌کنه و نهایتا 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: ۱۳۹۲/۰۱/۱۶ ۱۶:۴۶

من منظور شما را خوب متوجه میشم ولی حرفام یه بحث انتزاعی نیست چون پروژه عملی زیر دستم دارم که توی اون هم با پر کردن View Model کار میکنم.

مشکل از اینجا شروع میشه 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 دقیقا 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 مواقع هست که شما در پوسته برنامه به شکل دیگه ای از داده‌ها (متفاوت با اونچه در Model تعریف کردید و EF باهاش کار میکنه) نیاز دارید. مثلا 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, Update و Delete. 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 رووش اعمال میکنه (که به نظر من آقای خوشبخت هم به خوبی از کلمه Convert در لایه سرویس استفاده کردند.)

اما بحث اینکه ما در لایه مخزن روی EF یک سطح کپسوله میسازیم جای گفتگو داره هرچند من در اون مورد هم با وجد 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 ویـــو مدل تبدیل بشند و در این الگو این کار در لایه سرویس انجام میشه.

نویسنده: محسن
تاریخ: ۱۳۹۲/۰۱/۱۶ ۱۷:۱۰

- 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 میشه و اینکار اساسا اشتباه هست. چون هم شما رو محروم می‌کنه از قابلیت‌های پیشرفته ORM و هم ارزش افزوده‌ای رو 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 اومده این‌ها رو کپسوله کرده و لایه‌ای هست روی اون. اینکه ما مجددا یک پوسته روی این بکشیم حاصلی نداره بجز تکرار کد. عده‌ای عنوان می‌کنند که حاصل اینکار امکان تعویض ORM رو ممکن می‌کنه ولی این‌ها هم بعد از یک مدت تجربه با ORMهای مختلف به این نتیجه می‌رسند که ای بابا! حتی پیاده سازی LINQ این ORMها یکی نیست چه برسه به قابلیت‌های پیشرفته‌ای که در یکی هست در دوتای دیگر نیست (واقع بینی، بجای بحث تئوری محض).

- اینکه این تبدیلات (پر کردن ViewModel از روی مدل) هم می‌تونه و بهتره که (نه الزاما) در لایه سرویس انجام بشه، نتیجه مناسبی هست.

UI

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 لایه‌ی 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 سازنده‌ی کلاسهایی را که از الگوی Dependency Injection استفاده نموده اند و قطعا پارامتر ورودی آنها از جنس یک Interface می‌باشد را، با ایجاد شی مناسب مقداردهی می‌نماید.

به منظور استفاده از StructureMap در Visual Studio 2012 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

3. Installing 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 باشید که در این صورت به جای فضای نام SoCPatterns.Layered.WebUI از SoCPatterns.Layered.WinUI استفاده نمایید.

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. متد ConfigureStructureMap باید در همان لحظه ای که Application آغاز به کار می‌کند فراخوانی و اجرا شود. با توجه به نوع UI خود یکی از روالهای زیر را انجام دهید:

در WebUI :

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();
        }
    }
}
```

در WinUI :

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 Form1());
        }
    }
}
```

سپس برای طراحی رابط کاربری، با توجه به نوع UI خود یکی از روالهای زیر را انجام دهید:

در WebUI :

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"><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>
```

در WinUI :

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:

در WebUI :

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);
            }
        }
    }
}
```

در WinUI :

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 متد ObjectFactory.GetInstance مربوط به StructureMap ، 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 خواهم 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 Applicaion 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: ۱۳۹۲/۰۱/۰۳ ۱۴:۲۹

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 جزییات داره را کامل کنیم
آیا نمونه ای از این برنامه‌ها هست که این معماری یا معماری‌های مشابه را بسازه؟

نویسنده: شاهین کیاست
تاریخ: ۱۳۹۲/۰۱/۰۳ ۱۵:۳۱

اگر با T4 آشنایی داشته باشید بر اساس هر قالبی می‌توانید کد تولید کنید.

نویسنده: حامد
تاریخ: ۱۳۹۲/۰۱/۰۳ ۱۶:۳۶

متاسفانه آشنایی ندارم میشه یه توضیح مختصر بدین و یا منبع معرفی کنید

نویسنده: محسن
تاریخ: ۱۳۹۲/۰۱/۰۳ ۲۳:۵۹

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 خطایی رخ داده بسنده کنید. خطا رو مثلا توسط ELMAH لاگ کنید برای بررسی بعدی برنامه نویس.

نویسنده: شاهین کیاست
تاریخ: ۱۳۹۲/۰۱/۰۴ ۱:۲۰

http://codepanic.blogspot.com/2012/03/t4-enum.html

نویسنده: M.Q
تاریخ: ۱۳۹۲/۰۱/۰۴ ۲۲:۱۵

دوست عزیز غیر از ELMAH ابزار دیگری برای لاگ گیری از خطاها وجود دارد که قابل اعتماد باشد؟

همچنین اگر ابزاری جهت لاگ گیری از عملیات کاربران (CRUD => حالا R خیلی مهم نیست) می‌شناسید معرفی نمائید.

با سپاس

نویسنده: محسن
تاریخ: ۱۳۹۲/۰۱/۰۵ ۰:۳۳

متد auditFields مطرح شده در مطلب ردیابی اطلاعات این سایت برای مقصود شما مناسب است.

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۰۱/۱۲ ۱۴:۲۳

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 لطفا تصحیح بفرمایید.

نویسنده: علي
تاریخ: ۱۳۹۲/۰۱/۱۲ ۱۶:۳

مثال وب فرم هست. post back و page load داره.

نویسنده: شاهین کیاست
تاریخ: ۱۳۹۲/۰۱/۱۲ ۱۶:۴

اگر توجه کنید از الگوی MVP در Web Forms استفاده شده و نه در MVC.

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۰۱/۱۲ ۱۸:۳۰

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 ASP.NET Web form هست و در MVC می‌دونم که Page_Load و.. وجود نداره سوال من چیز دیگه بود دوستان

نویسنده: شاهین کیاست
تاریخ: ۱۳۹۲/۰۱/۱۲ ۱۸:۴۴

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 لطفا تصحیح بفرمایید.
با خواندن کامنت شما برداشت کردم شما تصور کردید کدهای 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 قبلا در مثال‌های آقای نصیری دیده ایم کافی است Service Layer در Controller مدل مناسب را تغذیه کند و برای View فراهم کند.

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۰۱/۱۲ ۱۹:۲۶

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 خوشبخت سوال کردم اما طاهرا هنوز وقت نکردن پاسخ بدن.

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 دوره ها است، که بحثی بر سر این معماری بشه.

نویسنده: شاهین کیاست
تاریخ: ۱۳۹۲/۰۱/۱۲ ۲۰:۴۶

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: ۱۳۹۲/۰۱/۱۲ ۲۱:۳

حرف شما کاملا متین هست

من قبلا معماری سه لایه کار می‌کردم، که نمونه اون توی همین سایت بخش پروژه ها گذاشتم، اما الان با EF , MVC کمی به مشکل بر خوردم و درست نتونستم تا حالا لایه‌های مورد نظر برای خودم در پروژه‌ها تفکیک کنم، این معماری به نظرم جالب اومد، خواستم که الگوی کار هم توی اون به کار ببرم که به مشکل بر خوردم (چون درک درستی از الگوی کار پیدا نکردم یا شایدم کلا دارم اشتباه می‌کنم). البته به قول شما شاید این معماری هنوز تکمیل نشده پروژه اش، در هر صورت از پاسخ‌های شما متشکرم.

نویسنده: شاهین کیاست
تاریخ: ۱۳۹۲/۰۱/۱۲ ۲۱:۷

خواهش می‌کنم.
فقط جهت یادآوری مثال روش آقای نصیری با پوشش MVC و EF قابل دریافت است.

نویسنده: ابوالفضل روشن ضمیر
تاریخ: ۱۳۹۲/۰۱/۱۷ ۱:۴۵

سلام
با تشکر فروان از شما ...
اگر امکان داره این مثال که در قالی یک پروژه نوشته شده برای دانلود قرار دهید ...تا بهتر بتوانیم برنامه را تجزیه و تحلیل کنیم ....ممنون

نویسنده: فرشید علی اکبری
تاریخ: ۱۳۹۲/۰۱/۱۹ ۱۵:۵۳

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 پروژه رو قدم به قدم بخونیم وطراحی 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 مراحلش حدود 45 دقیقه وقت گذاشتم ولی درصد یادگیریش خیلی بالاتر بود تا گرفتن فایل آماده...
درضمن لازمه بگم که بخاطر رفع شک وشبهه در سرعت پردازش وبالا اومدن اطلاعات، من تست این روش رو با تعداد 155 هزار رکورد انجام دادم که کمتر از سه ثانیه برام لود شد... باوجودیکه کامپوننت‌های دات نت بار مختلفی رو هم روی فرمم قرار دادم که بیشتر به اهمیت لود اطلاعاتم در پروژه و فرمهای واقعی پی ببرم.
سئوال اینکه :

به نظر شما ما می‌تونیم روی این لایه‌ها الگوی واحد کار رو هم ایجاد کنیم یا نه؟ اصلا ضرورتی داره ؟

نویسنده: علیرضا کیانی مقدم
تاریخ: ۱۳۹۲/۰۱/۲۸ ۱۲:۸

با تشکر از نویسنده مقاله و اهتمام ایشان به بررسی دقیق مفاهیم ،
از آنجا که flexible و reusable بودن برنامه‌ها را نمی‌توان نادیده گرفت تا آنجا که این تفکیک پذیری خود به مسئله ای بغرنج تبدیل نشده و تکرر داده‌ها و پاس دادن غیر ضرور آنها را موجب نشود تلاش در این باره مفید خواهد بود .
امروزه توسعه دهنده گان به سمت کم کردن لایه‌های فرسایشی و حذف پیچیدگی‌های غیر ضرور قدم بر می‌دارند. خلق عبارات لامبادا در دات نت و delegate ها نمونه هایی از تلاش بشر برنامه نویس در این باره است .

نویسنده: مسعود2
تاریخ: ۱۳۹۲/۰۲/۰۹ ۹:۱۲

سلام
Business Rule-1ها و validation-2ها در کجای این معماری اعمال میشوند؟

منظور از DomainService چیست؟

ممکنه منابع بیشتری معرفی نمایید؟
ممنون.

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۲/۰۹ ۱۶:۲۶

منبع برای مطالعه بیشتر

عنوان: CheckBoxList در ASP.NET MVC
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۱۰ ۲۰:۴۳:۰۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

ASP.NET MVC به همراه HtmlHelper توکاری جهت نمایش یک ChekBoxList 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 جدید را نیز به آن اضافه کنید. در ادامه، برای متد Index آن، یک View خالی را ایجاد نمائید. 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:

```
13        [HttpPost]
14        public ActionResult Index(string[] result)
15        {
16            return View();
17        }
18    }
```

result {string[2]}
[0] "value2"
[1] "value3"

بله. همانطور که ملاحظه می‌کنید، تمام عناصر ارسالی انتخاب شده که دارای نامی مشابه بوده‌اند، به یک آرایه قابل بایند 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 HtmlHelper سفارشی CheckBoxList استفاده کرد.

برای این منظور یک پوشه جدید را به نام app_code، به ریشه پروژه اضافه نمائید. سپس یک فایل خالی را به نام Helpers.cshtml 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، متد Helpers.CheckBoxList در تمام Viewهای برنامه در دسترس خواهد بود. در این متد، یک نام و لیستی از SelectListItemها دریافت می‌گردد. سپس به صورت خودکار یک CheckboxList را تولید خواهد کرد. برای دریافت مقادیر ارسالی آن به سرور هم باید مطابق متد GetTags تعریف شده در کنترلر Home عمل کرد. در اینجا Value عناصر انتخابی به صورت آرایه‌ای از رشته‌ها در دسترس خواهد بود.

روشی جامع‌تر

در آدرس زیر می‌توانید یک HtmlHelper بسیار جامع را جهت تولید CheckBoxList در ASP.NET MVC بیابید. 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: ۰۷:۵۸:۱۴ ۱۳۹۱/۰۲/۱۲

با سلام خدمت آقای نصیری عزیز
روز معلم و استاد را به شما تبریک می گویم
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 ناچار از این لغب استفاده نمی کنم
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 پروژه هایش، یکی در کتابش، یکی در تحقیق و غیره
پس تبریک این روز به شما حداقل کاری است که می توانیم انجام دهیم

نویسنده: وحید نصیری
تاریخ: ۰۸:۴۴:۱۸ ۱۳۹۱/۰۲/۱۲

سلام، ممنون. شما لطف دارید.
سلامت باشید و موفق

نویسنده: Naser Tahery
تاریخ: ۱۷:۰۱:۴۹ ۱۳۹۱/۰۲/۱۲

من تعریف نمیکنم چون میترسم خوشتون نیاد.
فقط همینقدر که امام علی فرمودند : "هر کس کلمه ای را به من بیاموزد مرا بنده ی خویش کرده است."
ولی من تا الان اینجا اندازه 30 تا کتاب مفید، مطلب یاد گرفتم.
روزت مبارک مهندس

نویسنده: سعید شیرزادیان
تاریخ: ۲۱:۰۴:۲۴ ۱۳۹۱/۰۲/۱۲

استاد نصیری عزیز خسته نباشید
زحمات بسیاری در رابطه با آموزش کشیدید متشکر. امکانش هست که مثالی از نحوه فراخوانی اطلاعات از دیتابیس بوسیله Entity Framework در mvc نیز بیاورید
متشکرم

نویسنده: reza
تاریخ: ۲۲:۴۶:۴۸ ۱۳۹۱/۰۲/۱۳

این دوست عزیز راست می گه بعضی ها نون شبی که می برن خونه از برکت مطالب شماست واقعا مطالبتون عالی و کاربردی است

نویسنده: محسن
تاریخ: ۱۶:۴۳ ۱۳۹۱/۰۴/۱۲

با سلام و خسته نباشید و تشکر از وبلاگ واقعا عالی و پر محتواتون
یه سوال از خدمتتون داشتم
من به این روشی که گفتین در پروژه 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 ممکنه چند نقش انتخاب بشه و اینو داخل پایگاه داده ثبت میکنم.
حالا سوال من اینجاست که وقتی میخام موقع ویرایش یک User نقش‌های اونو 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. لطفا 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 نهایتا 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 متدی یک new دارید. یعنی یک اتصال به بانک اطلاعاتی. این مورد در برنامه‌های وب می‌تونه 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 متد و n کلاسی که برای انجام یک کار مورد نیاز هستند.
در این مورد قبلا 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 گرید2 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 میگرددبه کنترلر از فروم‌های مشابه قبلا 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 مثلا یک ساعتی سعی در بازسازی مثال شما خواهد کرد. بعد سعی می‌کند تا خطای شما را به نحوی باز تولید کند و بعد هم مثلا 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 اصلا چه اطلاعاتی در برگه شبکه

این افزونه زمان ارسال به سرور لاگ شده. این را بررسی کنید و سپس با ساختار داده کنترلر خودتون مقایسه‌اش 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 HTML کار شما هم حتما result نام دارد (خصوصا اگر خودکار تولید می‌شود). بسته به نام واقعی عناصر انتخابی، نام این پارامتر متغیر خواهد بود. اما اصول نهایی یکی است.
- ضمنا روش دوم یافتن نام واقعی پارامترهای ارسالی، استفاده از روش قدیمی Request.Form در یک اکشن متد هست. این خاصیت دریافتی را دیباگ کنید تا به نام‌های واقعی برسید.

نویسنده: حسین فرهنگ
تاریخ: ۱۳۹۳/۱۲/۱۶ ۱۰:۵۶

این لینک پیام 404 رو میده https://github.com/devnoob/MVC3-Html.CheckBoxList-custom-extension

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۲/۱۶ ۱۱:۲۹

CheckBoxList(For) - A missing MVC extension

عنوان: RadioButtonList در ASP.NET MVC
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۲/۱۲ ۰۹:۲۷:۰۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

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. محتویات 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 متد HandleForm1 یک breakpoint قرار دهید. اکنون برنامه را اجرا کنید و فرم را به سرور ارسال نمائید. 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 نمائید.
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.

تهیه یک Templated helper سفارشی

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 Templated helper سفارشی را به نام EnumRadioButtonList، ایجاد کنیم تا در زمان فراخوانی متد Html.EditorForModel، به صورت خودکار enum تعریف شده را به صورت یک RadioButtonList نمایش دهد.

برای این منظور فایل جدید Views\Shared\EditorTemplates\EnumRadioButtonList.cshtml را به برنامه اضافه کنید. محتوای آن‌را 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 آن‌را 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.

نهایتا 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 متد Html.EditorForModel، به صورت خودکار تبدیل به یک RadioButtonList می‌شوند.

ج) متد زیر نیز همین کار را انجام می‌دهد:

```
@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.

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۳/۱۱/۰۴ ۲:۳۵

سلام؛ در صورتی که بخوایم این روش توی یک ویو مدل بکار ببریم روش کار چه جوری میشه؟
چون من اسم گروپ توی ویو مدل نوشتم نال برگردوند

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۱۱/۰۴ ۹:۵۵

من که کدی نمی‌بینم اینجا برای دیباگ و بررسی: آناتومی یک گزارش خطای خوب

ولی احتمالا نام خاصیت ViewModel شما با نام گروهی که تعریف کردید یکی نیست چون در مثال اول بین پارامتر string tech و کلاس زیر فرقی نیست:

```
public class SomeClass
{
  public string Tech {set; get;}
}
```

عنوان: **مدیریت محل اعمال Google analytics در ASP.NET MVC**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۳/۲۹ ۱۹:۵۸
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, MVC

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 واقعا 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 اسکریپتی را دریافت خواهید کرد که باید به انتهای تمام صفحات سایت خود اضافه نمائید 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؟
احتمالا 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 متد متناظری که آن‌را 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 اکشن متدهای آن اعمال شود؛ یا فقط به یک اکشن متد خاص که Viewایی خاص را رندر می‌کند.

نحوه پیاده سازی تشخیص وجود فیلتر Authorize را در یک View رندر شده، در متد کمکی زیر می‌توان 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 می‌توان به کنترلری که در حال رندر کردن View جاری است دسترسی یافت. اکنون که به کنترلر دسترسی داریم، به کمک Reflection، ویژگی‌ها یا Attributes آن‌را یافته و بررسی می‌کنیم که آیا دارای 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 منتسب 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 نوشتن 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 قبلی و ارسال‌های جدیدتر وجود داشت که میشد به راحتی به ترتیب ارسال پست، سایت رو مرور کرد.
در صورتی که این امکان گذاشته بشه، از مطالب بهتر میشه استفاده کرد

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۳/۲۹ ۲۱:۱۰

ممنون. 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 0 نظر داره ولی داخل خود پست که میشی میبنی 2 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 سایتت قرار بدم خیلی خوشحالم

ایکاش مانند گذشته فایل chm سایت رو نیز در اختیار مراجعه کننده‌ها 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 اولا سطح مطالب افت کنه و یا زیادی پراکنده بشه اگر برای نویسندگان سایت راهبری هایی انجام بشه شاید مطالب هماهنگ‌تر و مفیدتر و مورد نیازتر بشن متشکرم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۳/۲۹ ۲۳:۳

سلام؛ ممنون. برچسب‌های سایت کم نیست. لیست کامل آن‌ها در اینجا قابل مشاهده است:
http://www.dotnettips.info/search/label

نویسنده: مومن
تاریخ: ۱۳۹۱/۰۳/۳۰ ۸:۵۳

مبارک است
هر روز بهتر از دیروز!

نویسنده: مهدی
تاریخ: ۱۳۹۱/۰۳/۳۱ ۱۲:۳۹

واقعا عالی
عجب انرژیکی آقا وحید شما....

نویسنده: مجتبی حسینی
تاریخ: ۱۳۹۱/۰۳/۳۱ ۲۱:۲

جناب مهندس عزیز
مثل اینکه چند روزی هست که google analytic نیز ایران رو تحریم کرده چاره چیه؟جایگزین داره؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۳/۳۱ ۲۱:۲۸

مثل بقیه سرویس‌های آن با V.P.N کار می‌کنه. شما فقط نیاز به یک tracking id دارید. ثبت سایت و دریافت آن کار جمعا یک تا دو دقیقه است.

نویسنده: آریا
تاریخ: ۱۳۹۱/۰۴/۰۱ ۱۲:۴

ممنون بابت مطلب خوبتون

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۲/۰۵/۰۱ ۱۸:۱۶

با سلام. امکان دارد در مورد نحوه استخراج اطلاعات از این سرویس و نمایش در سایت کمی راهنمایی کنید؟ باتشکر.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۵/۰۱ ۱۸:۲۶

Google APIs Client Library for .NET

DotNET Client Library Developer's Guide

نویسنده: 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 ، نتیجه ای 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;
```

نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۰۲/۲۴ ۱۱:۲۰

- این روش منسوخ شده‌است و مربوط است به پروتکل‌های 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 پروتکل‌ها، در بسیاری از اوقات گوگل نیاز دارد تا شما توسط مرورگر سیستم جاری (در اینجا سرور وب)، یکبار به اکانت خود لاگین کنید.
- ga:pageviews با visits ارتباطی ندارد. فقط به این معنا است که صفحات سایت شما چندبار مشاهده شده‌اند و نه به معنای اینکه چندنفر آن‌هارا مشاهده کرده‌اند.

روش بهتر در اینجا

نویسنده: امیر
تاریخ: ۱۳۹۳/۰۳/۰۷ ۹:۳۱

با سلام و درود؛ یک سوال ؟ برای بدست آوردن کاربران Unique از session استفاده میکنیم و اما گوگل با استفاده از کدهای جاواسکریپت چطور میتونه 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: ۱۳۹۳/۰۳/۰۷ ۹:۳۸

در Google Analytics (نه راه حل‌های شخصی):
تعداد بار مراجعه کاربران: «sessions»
کاربران منحصربفرد مراجعه کننده به سایت: «users»
- تفاوت بین ga:sessions و ga:users در Google Analytics
- آنالیز محتوای کوکی‌ها utma و utmz + بررسی IP کاربر و تهیه یک چکیده از user-agent مرورگر و خیلی از مسایل ریز دیگر او برای تشخیص منحصربفرد بودن کاربر و نه فقط اینکه سشنی شروع شده، بنابراین کاربر جدید است.

نویسنده: سینا س
تاریخ: ۱۳۹۳/۱۲/۱۴ ۱:۱۹

سلام؛ میخواستم بدونم در صورتیکه قصد داشته باشیم آنالیز ترافیک بازدید سایت رو بدون استفاده از سرویس‌های مختلف (PersianStat, Google Analytic)، در وب سایت خودمون پیاده سازی کنیم چه روش هایی پیشنهاد میکنین؟

وب سایت بر پایه MVC 5 ساخته شده. روشی که خودم در نظر دارم، یک جدول در بانک اطلاعاتی هست که شامل یک سری فیلد مانند IP و User-Agent و تاریخ و ساعت و ...

به ازای هر کاربر یک رکورد اضافه بشه در دیتابیس. ولی به صورتی باشه که فقط یکبار اضافه بشه در یک مدت زمان محدود. یعنی به ازای هر Request که از یک کاربر مشخص ارسال میشه نیاد پشته سر هم هی رکورد درج کنه و به قول معروف با Refresh کردن هی رکورد ثبت بشه توی دیتابیس.

Session_Start در فایل Global گزینه‌ی خوبی هست برای اینکار ولی در وب سایت از Session استفاده نمیکنم و غیر فعالش کردم در وب کانفیگ، بنابراین جوابگو نیست.
ممنون میشم اگر بتونید راهنمایی کنید.

نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۱۲/۱۴ ۱:۳۱

- از رخداد Application_BeginRequest باید استفاده کنید.
- همچنین اگر سرور متعلق به خودتان است، روش لاگ‌های IIS ، امکاناتی مشابه google analytics را در اختیار شما قرار می‌دهد (^ و ^ و ^).

عنوان: **استفاده از HttpGet در ASP.NET MVC، آری یا خیر؟!**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۰۱ ۱۹:۵۷
آدرس: www.dotnettips.info
برچسب‌ها: ASP.Net, MVC

در ASP.NET MVC به کمک یک سری فیلتر می‌توان مشخص کرد که یک اکشن متد تنها به درخواست‌هایی از نوع Get پاسخ دهد، دیگری به درخواست‌هایی از نوع Post و الی آخر. عادت متداول من هم برای نمایش معمولی صفحات، استفاده از حالت HttpGet است که در شبکه‌های داخلی بدون مشکل کار می‌کند چون Bot ایی در این شبکه‌ها وجود ندارد و اگر باشد احتمالا 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 Botها بجای Get جهت دریافت اطلاعات فید، از Post هم استفاده می‌کنند! (مانند فیدبرنر گوگل مطابق لاگ‌های برنامه!) و در این حالت اگر متد 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 لاگ‌های خطای برنامه مواجه خواهیم شد.

بنابراین بهتر است که در سایت‌های عمومی، جهت ارائه اطلاعات و صفحات عمومی، از هیچ نوع فیلتری برای محدود کردن یک اکشن متد که کارش صرفا 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، بی‌جهت آن‌را محدود نکرده باشیم.

عنوان: **اثر وجود سشن بر پردازش موازی در ASP.NET**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۰۳ ۱۰:۱۳
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, MVC, Parallel Extensions

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+نوشتنی 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 نمائید 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 متدی (و یا تمام اکشن متدهای یک کنترلر) که برای درخواست‌های Ajax طراحی شده، خاموش گردیده است تا درخواست‌های غیرهمزمان Ajaxایی بتوانند حداکثر استفاده را از پردازش موازی خودکار کنند.
نکته دیگر اینکه اگر نیاز به استفاده از سشن در یک اکشن متد وجود 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 زیادی روی سرور ایجاد نمیکنه. تا چه تعداد کاربر رو 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 می‌دارم تا برای کارهای مختلف هربار 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 خواهید رسید.

نویسنده: علی
تاریخ: ۱۳۹۱/۱۰/۲۸ ۲۳:۳۰

ممنون، آیا اطلاعات دیگری را هم می‌شود با این روش ذخیره کرد، مثلا نام کاربر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۰/۲۸ ۲۳:۳۸

کاری که در اینجا انجام میشه 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

ممنون بابت راهنمایی خوبتون

نویسنده: رضا گرمارودی
تاریخ: ۱۳۹۲/۱۱/۱۹ ۱۱:۱۰

سلام من واقعا متوجه نحوه استفاده یا عدم استفاده از Sessionها نشدم!
در Asp.net برای نگهداری یکسری از اطلاعات کاربر و یا هرگونه اطلاعات مربوط به یک کاربر خاص از Sessionها 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 درعوض در هر درخواستی از آن استفاده شود و یا خیر هربار این اطلاعات واکشی شود (البته بدون درنظر گرفتن سطح دوم کش).
همچنین Sessionها در Mvc با Session در Asp.net چه تفاوت هایی دارد؟
ممنون میشم ابهام بنده را برطرف بفرمایید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۹ ۱۱:۳۰

- در سایت جاری ماژول سشن در وب کانفیگ کلا حذف شده. در هیچ‌جایی 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 رمزنگاری شده‌است؛ چه برای نگهداری اطلاعات کاربر لاگین شده به سیستم و چه برای نگهداری اطلاعات نقش‌های منتسب به او.
- تفاوتی ندارند. یک هسته‌ی اصلی ASP.NET وجود دارد که حداقل دو فریم ورک وب فرمز و MVC از آن استفاده می‌کنند.

نویسنده: همراز
تاریخ: ۱۳۹۳/۰۹/۱۱ ۱۱:۰

سلام؛ اگر بخواهیم از سشن استفاده نکنیم، برای نگهداری اطلاعات کسی که لاگین کرده چه راهی وجود داره؟ مثلاً UserId یا LoginId که با اون بشه به UserId رسید.

نویسنده: 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.

عنوان: **استفاده از دکمه‌های CSS توئیتر در ASP.NET MVC**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۰۷ ۱۲:۴۴
آدرس: www.dotnettips.info
برچسب‌ها: ASP.Net, MVC

یک سری نکته ریز را جهت بهبود ظاهر برنامه‌های وب می‌توان درنظر داشت؛ برای مثال:
مجموعه‌ی 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 class آن‌را مثلا 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()">ورود به سیستم</span>
```

jQuery که جزء پیش فرض برنامه‌های ASP.NET MVC است. متد submit را هم می‌توان در اینجا در صورت کلیک بر روی یک span فراخوانی کرد. نکته مهم آن ذکر Id فرم جاری است که مثلا 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 میذاره. 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

عنوان: **نحوه صحیح تولید Url در ASP.NET MVC**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۰۸ ۱۰:۱۶
آدرس: www.dotnettips.info
برچسب‌ها: ASP.Net, MVC

کار متد کمکی Url.Action ایجاد یک Url بر اساس تعاریف مسیریابی برنامه است. البته متد کمکی مشابهی نیز مانند Html.ActionLink در ASP.NET MVC وجود دارد که کار آن تولید یک لینک قابل کلیک است؛ اما ممکن است در حالتی 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 Ajaxایی 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 ایی 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 در 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 کنترلرها، وهله 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 کنترلرها 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 نباید Urlها را دستی تولید کرد. در Html Helperهای سفارشی نیز می‌توان به کمک متد UrlHelper.GenerateUrl، به همان امکانات Url.Action 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

نویسنده: محسن
تاریخ: ۱۳۹۱/۰۴/۱۰ ۱۵:۶

سلام و خسته نباشید خدمت شما آقای نصیری. وبلاگتون واقعا مفید و عالیه
ممنون از زحماتتون بابت مطالبی که به اشتراک میگذارید
یه سوال از خدمتتون 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 کاملا با Jquery کار کردم و Insert , Update , Delete کلا توسط Jquery انجام میشه. اما توی یکی از Viewهای 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 کارمو انجام میدم. اما برای "ایجاد" یک فرم از نوع فرم‌های 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 مستقیما داخل TextBoxها قرار بگیره. البته توسط Jquery اومدم تک تک textboxها رو به صورت زیر مقدار دادم. حالا نمیدونم کارم درسته یا روش بهتری هست واسه این کار. اگر نیاز هست تا سورس برنامه رو واستون 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.LastLoginDate);
                    $("#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"); }
        });
    }
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۱۰ ۱۶:۵۹

فرم ادیت رو به صورت strongly typed از نوع یک partial view مستقل درست کنید.
سیس در کنترلر مرتبط، یک اکشن متد 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 نهایتا 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.

+

لطفا 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: ۱۳۹۱/۰۵/۲۰ ۲:۱۸

سلام جناب نصیری ممنون از مطالب مفیدتون.
سوال:
همون طور که مرسوم هست معمولا 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 داره.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۲۰ ۸:۳۷

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 آن‌را 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: ۱۳۹۲/۰۸/۲۶ 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.Schem)
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۸/۲۶ ۱۴:۲۹

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.

عنوان: **استفاده از OpenID در وب سایت جهت احراز هویت کاربران**
نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۱/۰۴/۰۸ ۲۱:۲
آدرس: www.dotnettips.info
برچسب‌ها: ASP.Net, C#, MVC, Security, OpenID

قبلا 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.
1- ابتدا در ویژوال استودیو یا هر نرم افزار دیگر یک پروژه MVC ایجاد نمایید.

2- نوع Internet Application و برای View Engine سایت Razor را انتخاب نمایید.

3- کتابخانه DotNetOpenId سورس باز را می‌توانید مستقیما 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 به نام LogOn یک 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
                indentity. 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>
    </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 ودانلود نمونه کدهای آماده می‌توانید به لینک‌های (^ و ^ و ^ و ^ و ^ و ^ و ^) مراجعه کنید.
کد کامل پروژه را می‌توانید از اینجا دانلود نمایید.

منبع

نظرات خوانندگان

نویسنده: ahmadalli
تاریخ: ۱۳۹۱/۰۴/۰۸ ۲۳:۱۲

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: ۱۳۹۱/۰۴/۰۹ ۱:۲۵

اگه میشه کد برنامه را ضمیمه کنید...

نویسنده: صابر فتح اللهی
تاریخ: ۱۳۹۱/۰۴/۰۹ ۲:۴

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 اللهی
تاریخ: ۱۳۹۱/۰۴/۰۹ ۲:۲۱

اصلاح شد

نویسنده: saleh
تاریخ: ۱۳۹۱/۰۴/۱۷ ۰:۳۳

فرضاً اگه شخصی که با OID لوگین کرده بخواد 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: ۱۳۹۱/۰۴/۱۷ ۱۱:۴۴

بحث فوروم نوشته شده شما با طراحی یک سیستم جدید (که در اینجا مد نظر است) متفاوت است. سورس فوق را مطالعه کنید تا با نحوه یکپارچگی آن با سیستم membership دات نت آشنا شوید.

نویسنده: احمدعلی شفیعی
تاریخ: ۱۳۹۱/۰۴/۱۸ ۱۲:۳۸

ممنونم

نویسنده: کامران
تاریخ: ۱۳۹۲/۰۲/۱۲ ۲۳:۲۱

سلام دوست عزیز.
قربان برای ASP.NET Webforms هم میتونید مقاله ای رو آماده کنید؟

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

ممنون 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 متاسفانه

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۰۲/۱۳ ۰:۲۵

سلام
دوست گلم یه مثال هست گذاشتم توی این آدرس

Google OpenID Authentication در ASP.NET با استفاده از DotNetOpenAuth

عنوان: **متدهای کمکی مفید در پروژه های asp.net mvc**
نویسنده: امیرحسین جلوداری
تاریخ: ۱۳۹۱/۰۴/۰۹ ۱۶:۱۶
آدرس: www.dotnettips.info
گروه‌ها: MVC

ابتدا در پروژه‌ی 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 ی 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- خلاصه‌ی مطلب با استفاده از کتابخانه ی Html Agility Pack (تعداد کلمات از کلمه‌ی اول یک متن به اندازه‌ی 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 Errorها 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 پروژه‌های 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 seo friendly کردن URL 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

ببخشید 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 برمیگردونه

```
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

تاریخ: ۱۳۹۱/۰۴/۱۷ ۲۳:۱

بهتره این مباحث رو در انجمن یا mailing list خودش بررسی کنید.
زمانیکه شما نودهای radio رو دارید بقیه‌اش چیزی شبیه به این خواهد بود

```
string isChecked = element.GetAttributeValue("checked", "unchecked");
```

نویسنده: میثم شریفی
تاریخ: ۱۳۹۱/۰۴/۲۵ ۲۰:۱۱

```
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: ۱۳۹۱/۰۵/۱۶ ۲۱:۵۴

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 متغییری 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 اتفاده 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: ۱۳۹۱/۰۵/۱۶ ۲۲:۴۹

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 مثلا Razor استفاده کرد اما این کدها در زمان اجرا به صورت جاوا اسکریپتی پردازش و رندر می‌شوند و نه به فرمت مثلا سی شارپ. 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 فراخوانی آن باید داخل کدهای Ajax جاوا اسکریپتی یک View باشد (و نه خارج از آن چون context دیگر JavaScript نخواهد بود). همچنین باید درنظر داشت که Html.FarsiDateAndTime 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: ۱۳۹۱/۰۵/۱۶ ۲۳:۱۴

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 خطا میدهد

راه بهتری هست که قبل از Return عمل تبدیل تاریخ را انجام دهم و سپس return Json(data
JsonRequestBehavior.AllowGet, را انجام دهم.
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: ۱۳۹۱/۰۵/۱۶ ۲۳:۲۵

چه خطایی می‌ده؟ چون من الان استفاده کردم و خطایی 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.
```

مربوط است به نال بودن یکی از مقادیر تاریخ ثبت شده در بانک اطلاعاتی. خاصیت را در سمت کدهای خود nullable تعریف کنید.

نویسنده: hasani

تاریخ: ۱۳۹۱/۰۵/۱۶ ۲۳:۴۱

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 نگارش‌های قبل از آن این روش جواب می‌دهد:

```
db.table.Where(...)
         .Select(p =>
                    new
                    {
                       p.Date1,
                       p.Comment,
                       p.Cost,
                       p.UserId,
                     })
                  .AsEnumerable() // مهم
                  .Select(...در اینجا مجاز هستید از هر نوع تابعی استفاده کنید..)
                  .ToList();
```

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 راه‌حل مناسبی نیست؟

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۸/۰۵ ۱۲:۳۱

تمام این‌ها در پشت صحنه از همین فضای نام استفاده می‌کنند. فقط کمی ساده‌ترش کردن.

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۳/۱۰/۰۸ ۲۱:۲۲

اما این regex کدهای ویژه حذف نمیکنه مثل © یا مثلا nbsp , ...

نویسنده: امیرحسین جلوداری
تاریخ: ۱۳۹۳/۱۰/۰۹ ۰:۱۴

بنده دستی این موارد رو حذف کردم (حاصل از مشاهده در پروژه‌های مختلف) :

```
public static string RemoveAllHtmlTags(this string text)
        {
            var withoutHtml = String.IsNullOrEmpty(text) ?
                          String.Empty :
                          Regex.Replace(text, @"<(.|\n)*?>", String.Empty);



            withoutHtml = withoutHtml.Replace(" ", " ");
            withoutHtml = withoutHtml.Replace("&zwnj;", " ");
            withoutHtml = withoutHtml.Replace("&quot;", " ");
            withoutHtml = withoutHtml.Replace("amp;", "");
            withoutHtml = withoutHtml.Replace("&laquo;", "«");
            withoutHtml = withoutHtml.Replace("&raquo;", "»");

            return withoutHtml;
        }
```

عنوان: T4MVC : یکی از الزامات مدیریت پروژه‌های ASP.NET MVC
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۱۱ ۹:۴۵
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, MVC, T4MVC

تعریف ActionLink زیر را درنظر بگیرید:

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

پارامترهای دوم و سوم آن که به نام‌های یک اکشن متد 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 متد را در برنامه تغییر داد. به این ترتیب تمام ActionLink هایی که در برنامه به این کنترلر اشاره می‌کردند از کار می‌افتند (تکرار رشته‌ها به علاوه refactoring friendly نبودن آن‌ها).
ب) برای نوشتن رشته‌ها intellisense کارآیی ندارد.
ج) امکان بروز اشتباهات تایپی در این بین بسیار زیاد است.

راه حل متداولی که برای حل این نوع مشکلات وجود دارد، تعریف یک کلاس عمومی و معرفی رشته‌ها به صورت فیلدهایی ثابت در آن‌ها می‌باشند و سپس استفاده از این فیلدها 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!
خوشبختانه 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 David Ebbo از اعضای تیم ASP.NET MVC تهیه شده است.

پس از دریافت پروژه، تنها به دو فایل زیر از آن نیاز داریم:
T4MVC.tt و T4MVC.tt.settings.t4

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). بلافاصله این فایل‌های t4 وارد عمل شده و کلاس‌های کنترلرها، 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 معادل رشته‌ها است. همانطور که ملاحظه می‌کنید، یک سری overload را هم به متدهای پیش فرض ASP.NET MVC اضافه کرده است و حتی بجای معرفی رشته معادل 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 متداول 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 مستقیما 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 بپذیرد. 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 متداول).

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 پوشه shared می‌توان از MVC.Shared.Views 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) پوشه‌های Content و Scripts هم دسترسی یافت و حتی این موارد را نیز refactor کرد:

```
<img src="@Links.Content.Images.arrow_right_png" alt="arrow" />
<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 متد Index یک کنترلر از نوع strongly typed است و هم امکان تعریف پارامترهای آن به نحو منطقی‌تری فراهم شده است (متد Comments.Index واقعا 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؛ وگرنه ثوابتی را که ایجاد می‌کند کامپایل شده و در زمان اجرا سرباری را به برنامه اضافه نمی‌کنند). بنابراین اگر کلاسی به پروژه اضافه شده است، کامپایل کردن آن‌را فراموش نکنید.

-اگر تغییری در فایل‌های View، در تعداد و نام آن‌ها صورت گرفت، روی فایل T4MVC.tt کلیک راست کرده و گزینه‌ی اجرای آن‌را انتخاب کنید. پس از این‌کار، مجددا 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 آن‌را مقدار دهی کرد.

- overloadهای 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 قبلا 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 بلافاصله 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 چ جوری باید عملل کرد من به صورت زیر میدم syntax error میده

```
@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 LaodWordInfo(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();
```

```
                    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 بده.
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: ۱۳۹۱/۰۴/۱۱ ۲۱:۲۴

در کد فوق که از T4MVC استفاده نشده ولی نحوه تعریف Url شما اشتباه است. قبلا 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 section به نام مثلا JavaScript درست کنید. در اینجا می‌شود داخل کدهای جاوا اسکریپتی هم از Razor استفاده کرد. سپس این section را در layout در قسمت header صفحه include کنید (در قسمت 14 سری MVC این سایت در مورد این تکنیک کاملا توضیح دادم).

نویسنده: رضا.ب
تاریخ: ۱۳۹۱/۰۴/۱۲ ۲:۴۱

نمونه‌های شبیه این باز هم بودند که مسئله‌ای که سالهاست دات نت با Reflection قابل حل کرده. ولی بازم تو جدیدترین تکنولوژی‌ها از مقادیر رشته‌ای استفاده میکنه. آیا هدف خاصی پشت این منطق هست؟ یا صرفا بی‌توجهی و قصور تیم توسعه هست؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۱۲ ۸:۳۶

Reflection اندکی کند است. در حین توسعه برنامه اگر ویژوال استودیو مرتبا بخواهد کوئری‌های مبتنی بر reflection از اسمبلی‌های شما بگیرد، سرعت کار شما کم خواهد شد و نهایتا ممکن است سبب ناپایداری آن هم بشود (اگر پروژه جاری بزرگ باشد).

نویسنده: افشار محبی
تاریخ: ۱۳۹۱/۰۴/۱۲ ۲۳:۷

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: ۱۳۹۱/۰۶/۱۰ ۱۷:۱۴

با سلام و خسته نباشید خدمت دوستان
آقا من نتونستم این فایل‌ها رو دانلود کنم، خواهش میکنم یه راهی بگین تا منم بتونم دانلودشون کنم.
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۱۰ ۱۸:۴۷

از اینجا دریافت کنید.

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۱/۰۸/۲۰ ۸:۵۹

مهندس سلام.
من موقع اضافه کردن T4mvc از طریق نوگت تحت mvc4 فایلی که T4MVC.tt.settings.t4 نام داشته باشد را نمی‌بینم. کجای کارم ایراد دارد؟ با تشکر.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۸/۲۰ ۹:۱۰

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 compatability as long as setting types
do not change.
```

نام جدید آن **xml**. T4MVC.tt.settings شده.
+ 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: ۱۳۹۱/۰۸/۲۰ ۹:۱۹

سپاس مهندس.

نویسنده: SMF
تاریخ: ۱۳۹۱/۱۲/۱۰ ۱۰:۳۲

با سلام

من تو V.S. 2012 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 برخوردن، بخوره ...

نویسنده: cooli
تاریخ: ۱۳۹۱/۱۲/۱۷ ۱۲:۵۰

دو سوال
1 - در ورژنی که بنده از NuGet نصب کردم (3.5.1) ، T4MVCExtensions هم نصب گردید ، کار T4MVCExtensions چیست ؟
2 - بعد از اجرا ، تمام کلاس‌های Controllerهای پروژه من partial شدند و تمام Actionهای من virtual ، چرا ؟ آیا مشکلی در آینده برای Controllerها و Actionها پیش نخواهد آمد ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۱۷ ۱۲:۵۶

- یک سری متد کمکی اضافی برای کار با T4MVC که به صورت پیش فرض در MVC وجود ندارد؛ خصوصا 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. مثلا زمانی که می‌نویسند View.MyPartialViewName، نیاز است کنترلر جاری partial تعریف شود تا به رشته یاد شده که توسط T4MVC اضافه می‌شود دسترسی یافت.
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 حدس‌زدن و سعی و خطا را توسط موتور View جاری برای یافتن منبع مورد نظر دارد.

نویسنده: امیر
تاریخ: ۱۳۹۱/۱۲/۲۰ ۹:۱۶

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 اصلا نمیشناسه

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۲۰ ۹:۱۹

run custom tool رو اجرا کردید؟

نویسنده: مولانا
تاریخ: ۱۳۹۲/۰۱/۲۶ ۱۱:۵۹

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 تعریف 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; }
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۱/۲۶ ۱۲:۴۰

از T4ResX استفاده کنید.

نویسنده: مولانا
تاریخ: ۱۳۹۲/۰۱/۳۱ ۱۶:۱۸

با سلام.
من پوشه ای با عنوان SiteContent در root پروژه دارم. چگونه می‌توانم از طریق T4MVC به آن دسترسی داشته باشم. درحالت عادی وقتی Links را تایپ میکنم فقط منابع را برای پوشه Content و Scripts شناسایی میکند. باتشکر.

نویسنده: 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>
```

نویسنده: علی
تاریخ: ۱۳۹۲/۰۴/۱۵ ۱۹:۴۶

با سلام بنده 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.GetOrCreateFileInfo(String fname, String text)
   at Microsoft.XmlEditor.Package.GetOrCreateFileInfo(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
```

مواجه شدم t4mvc را uninstall کردم دوباره نصب کردم موقع generate دوباره همین پیغام را میدم. لطفا یکی به من کمک کنه.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 کنترلرها یا اضافه کردن یک کنترلر جدید حتما باید run custom tool اجرا کنم...می خواستم بدنم آیا راهی نیست که بشه این قضیه خودکار انجام بشه ؟ (مثلا بعد از Build پروژه)

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 اکشن متدها نباشند. تغییرات در سایر لایه‌ها باشند. در این حالت اصلا 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 می کنم 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؟ لطفا راهنمایی کنید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۵/۲۶ ۰:۲۶

کل کاری که T4MVC انجام می‌ده، اضافه کردن یک سری متد الحاقی جدید است به پروژه به علاوه تبدیل یک سری از مسیرها به معادل‌های constant آن و اندکی تغییر در متدهای کنترلرها برای سازگار شدن با متدهای الحاقی جدید. بنابراین حاصل این عملیات به صورت کامپایل شده در پوشه bin پروژه شما موجود است. این پوشه را کلا بر روی سرور ارسال کنید.

نویسنده: رضا منصوری
تاریخ: ۱۳۹۲/۰۸/۲۲ ۰:۴۸

«من قبلا چطور با ASP.NET MVC کار می‌کردم؟!»

نویسنده: رویا حیدری
تاریخ: ۱۳۹۳/۰۳/۰۳ ۱۷:۱۶

من در پروژم Controller ها رو در یک لایه و Viewها 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 میشه ساخته نمیشه و Error میده! (آنهایی که RedirectToAction هستند به درستی ساخته می‌شه)

```
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: ۱۳۹۳/۰۳/۰۳ ۱۸:۴

- ViewsRootFolder داخل فایل تنظیمات آن، قابل تنظیم است.
- همچنین بحثی هم در اینجا شده با یک چنین راه حلی

نویسنده: احمد ولی پور
تاریخ: ۱۳۹۳/۰۷/۱۵ ۰:۴۲

من سه تا فایل T4MVC.tt و T4MVC.tt.settings.xml و حتی T4MVC.tt.hooks.t4 رو تو یه پوشه تو سطح اول میذازم ولی همش خطای زیر رو 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 ۱:۱۸

تنظیمات فایل‌های کانفیگ برنامه شما (bindingRedirect ها) صحیح نیست یا ناقص است. اطلاعات بیشتر

نویسنده: reza110
تاریخ: ۱۳۹۳/۰۸/۲۸ ۱۵:۴۳

با سلام
T4MVC یکسری تغییرات در برنامه ما ایجاد می‌کنه از جمله مثلا یک کانستراکتور برای کنترلر Home اضافه می‌کنه و من چگونه باید کدی مثل کد زیر رو داخل کنترل 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 ۱۸:۱۳

- روش « poor man's dependency injection » اساسا زیر سؤال هست. روش صحیح تزریق وابستگی‌ها در برنامه‌های ASP.NET MVC در اینجا بحث شده‌است.
- اگر نمی‌خواهید T4MVC سازنده‌ی پیش فرض ایجاد کند، فایل‌های T4 آن‌را ویرایش کنید .

عنوان: **مقدمه ای بر AutoMapper**
نویسنده: محمد صاحب
تاریخ: ۱۳۹۱/۰۴/۱۳ ۱۷:۲۲
آدرس: www.dotnettips.info
برچسب‌ها: ASP.Net MVC, AutoMapper, ASP.Net, MVC

AutoMapper 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 Model 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 بعنوان Model در View ی مورد نظر باید شی UserViewModel رو با اطلاعات شی User مقدار دهی کنیم مثلا 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 رو تعریف میکنم.یک روش ابداعی تعریف نگاشت‌ها در یک کلاس استاتیک و فراخوانی اون تو متد Application_Start هست.

```
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 متد 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 میبنید با نوشتن 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 مدتیه معطل همین مپینگ‌ها به صورت دستی هستم

نویسنده: سروش ترک زاده
تاریخ: ۱۳۹۱/۰۴/۱۳ ۱۷:۳۳

سلام
ممنون،
راه حل ساده واقعا لذت بخش هست...

نویسنده: daniel
تاریخ: ۱۳۹۱/۰۴/۱۳ ۲۱:۱۳

تو Compact Framwork از valueinjecter باید استفاده کرد.

نویسنده: امیرحسین جلوداری
تاریخ: ۱۳۹۱/۰۴/۱۳ ۲۳:۲

خیلی ممنونم ... میشه این سمپل رو ضمیمه کنید؟!

نویسنده: رضا.ب
تاریخ: ۱۳۹۱/۰۴/۱۳ ۲۳:۴۶

تشکر بابت معرفی.
کاربرد این Auto Mapper در اینجور نگاشت‌ها معنی میده فقط؟ یعنی نگاشت یه شیِ جزء به شیِ کل؟ یا مثلا کاربردهایی مثل ORM هم میتواند داشته باشد؟

نویسنده: محمد صاحب
تاریخ: ۱۳۹۱/۰۴/۱۴ ۹:۲۱

یکی از مهمترین کاربردهاش نگاشت از یک مدل بزرگ و پیچیده به یک مدل ساده هست که بهش Flattening میگن و کاربردهای دیگه مثلا Projection و... که همنطور 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

سمپل خاصی نیست کدها رو از یک پروژه کشیدم بیرون...
تنها قسمتی که تو این پست نذاشتم(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
ممنون 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 کمکی استفاده کن

نویسنده: hossein101211
تاریخ: ۱۳۹۲/۰۴/۱۰ ۱۳:۵۸

با سلام میخواستم ببینم آیا امکان استفاده از automapper بدین صورت هم وجود داره؟

```
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

نویسنده: محمد صاحب
تاریخ: ۱۳۹۲/۰۴/۱۰ ۱۴:۳۰

اگه q از نوع news هست که نیاز ی به AutoMapper نداری !

```
AutoMapper.Mapper.Map(news,q);
```

این خط مشکلی نداره

نویسنده: hossein101211
تاریخ: ۱۳۹۲/۰۴/۱۰ ۱۵:۵

من میخوام محتویات news رو (که توسط model binding پر شده) رو در سطری 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

نویسنده: محمد صاحب
تاریخ: ۱۳۹۲/۰۴/۱۱ ۱۱:۲۷

اگه نوع هاشون متفاوته بله باید Mapper.CreateMap رو تعریف کنی.

عنوان: **بهبود سرعت نمایش صفحات در ASP.NET MVC با حذف View Engines اضافی**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۱۳ ۲۳:۴۳
آدرس: www.dotnettips.info
برچسب‌ها: ASP.Net, MVC, ASP.Net MVC

در ASP.NET MVC امکان استفاده از چند View Engine به صورت همزمان وجود دارد و همچنین هربار که قرار است Viewایی رندر شود، از تمام این‌ها تا یافتن موتور مناسب نمایش View جاری کوئری می‌گیرد. بدیهی است هرچقدر تعداد موتورهای ثبت شده در اینجا بیشتر باشند، زمان بیشتری نیز برای یافتن موتور نمایشی مناسب صرف خواهد شد؛ خصوصا 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 صرفا 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 حتما دقت داشته باشید که نسخه MVC آن باید نصب شود تا برگه‌های Routing و View آن 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.

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 درگیر نمایش صفحه جاری بوده‌اند، مشخص شوند:

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:

/

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	RazorViewEngine	True	False	Not Found In: RazorViewEngine cache
2	Index		False	RazorViewEngine	False	True	Key / Value: ViewData --, Model --, TempData --

هم تعداد سعی و خطاها کمتر شده و هم تعداد فایل‌هایی که بررسی شده است به حداقل رسیده (برای مثال در حالتیکه موتور WebForms فعال باشد، چهار فایل با پسوندهای مختلف در مکان‌های پیش فرض نیز حتما جستجو خواهند شد).

نظرات خوانندگان

نویسنده: شهروز جعفری
تاریخ: ۱۳۹۱/۰۴/۱۳ ۲۳:۵۸

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: ۱۳۹۱/۰۴/۱۴ ۰:۱

خیر. همزمانی در اینجا به معنای امکان استفاده همزمان از فایل‌های مستقل aspx و cshtml یا vbhtml است.

نویسنده: شهروز جعفری
تاریخ: ۱۳۹۱/۰۴/۱۴ ۰:۱۹

یه سوال نامرتبط Ninject چیه؟کاره همون structuremap میکنه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۱۴ ۰:۲۷

بله. از این نوع کتابخانه‌ها زیاد است. Ninject، StructureMap، AutoFac و غیره. خود مایکروسافت هم یک نمونه دیگر به نام Unity دارد.

عنوان: **محدود کردن کاربرها به آپلود فایل‌هایی خاص در ASP.NET MVC**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۱۶ ۰:۱
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, MVC, Security, ASP.Net MVC

یک ادیتور آنلاین را تصور کنید که کاربران در قسمت ارسال تصویر آن قرار است فقط فایل‌های png، jpg و gif ارسال کنند و نه مثلا فایل test.aspx و موارد مشابه. در اینجا برای محدود کردن نوع فایل‌های آپلود شده می‌توان از فیلترهای سفارشی ASP.NET MVC 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 متد OnActionExecuting آن، توسط filterContext.HttpContext.Request.Files می‌توان به کلیه فایل‌های درحال 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: ۱۳۹۱/۰۴/۱۷ ۱۴:۵

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 استثنایی صادر شود. حالا برای فایل‌ها با فرمت‌های دیگه چیکار کنیم؟ (مثلا ممکن است بخواهیم به کاربر اجازه ارسال فایل‌های zip و یا rar بدهیم و مواردی از این دست).

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۱۷ ۱۴:۱۶

برای بررسی محتوایی خاص، نیاز به parser مخصوص این نوع فایل‌ها است تا بتواند بررسی کند محتوای دریافتی معتبر است یا نه و عموما کسی محتوای این نوع فایل‌ها را بررسی نمی‌کند. بنابراین بررسی پسوند در اکثر موارد کافی است.
مشکل این نیست که فایل rar واقعا rar است یا نه. مشکل این است که این فایل ارسالی، قابلیت اجرا را از طریق فراخوانی آدرس آن در مرورگر، نداشته باشد. بحث بررسی پسوند هم به همین دلیل است. فایل aspx را می‌شود از طریق فراخوانی در مرورگر بر روی سرور اجرا کرد ولی فایلی با پسوند rar این قابلیت را ندارد.

نویسنده: شاهین کیاست
تاریخ: ۱۳۹۱/۰۵/۰۹ ۲:۱

ممنون ، استثنایی که صادر می‌شود را چطور میشه ؟ در بدنه‌ی Action مربوطه مدیریت کرد ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۰۹ ۸:۲۳

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 رخ داده است.
ولی در کل می‌شود IExceptionFilter را نیز پیاده سازی و مدیریت کرد:

```
public class CustomFilter : FilterAttribute, IExceptionFilter
{
    public void OnException(ExceptionContext filterContext)
    {
```

نویسنده: رضوی
تاریخ: ۱۳۹۱/۰۵/۲۸ ۸:۵۳

سلام

چند نکته تکمیلی برای امنیت بیشتر
1- بهتر است که فایل روی دیتابیس ذخیره شود، در غیر اینصورت نام فایل و پسوند توسط سیستم تعیین شود.
2- نام و پسوند ارسالی در نظر گرفته نشود.
3- 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>
```

عنوان: **ارسال فایل در ASP.NET MVC و اعتبار سنجی سمت کاربر**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۲۳ ۱۸:۱۶
آدرس: www.dotnettips.info
برچسب‌ها: ASP.Net, MVC, ASP.Net MVC

پیشنیازها:
- نحوه ارسال فایل به سرور توسط ASP.NET MVC
- نحوه اعتبار سنجی سمت سرور ارسال فایل‌ها

در ASP.NET MVC برای آپلود فایل‌ها عموما 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 مثلا «لطفا 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="لطفا 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 خصوصا در طراحی html helperهای 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 آن‌را توسط یک attribute به سادگی به یک مدل خاص انتساب داد).
در این حالت تنظیم data-val و data-val-required به کمک anonymously typed objects 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

نویسنده: 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.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۱۷ ۱۸:۵۴

بله. مرورگرهایی که XMLHttpRequest Level 2 را پشتیبانی کنند (از IE 10 به بعد مثلا)، امکان Ajax upload توکار به همراه گزارش درصد آپلود را بدون نیاز به فلش یا سیلورلایت، دارند. یک نمونه پیاده سازی آن

عنوان: **فعال سازی قسمت ارسال فایل و تصویر ویرایشگر آنلاین RedActor در ASP.NET MVC**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۰۳ ۲۳:۲۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, ASP.Net MVC, jQuery

در سایت جاری از ویرایشگر آنلاین RedActor استفاده شده و کار کردن با آن هم بسیار ساده است:
یک TextArea ساده را به صفحه اضافه کرده و این افزونه جی‌کوئری را بر روی آن اجرا می‌کنید. به این ترتیب TextArea به 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 مثال‌های 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 fileUpload، imageUpload و linkFileUpload مقدار دهی شوند.
همانطور که ملاحظه می‌کنید از T4MVC برای م شخص سازی مسیرها استفاده شده. برای مثال
MVC.RedactorUpload.ImageUpload به این معنا است که در کنترلری به نام RedactorUpload، اکشن متدی به نام ImageUpload پذیرای ارسال فایل ادیتور خواهد بود و به همین ترتیب در مورد سایر پارامترها.
RedactorUploadController هم ساختار بسیار ساده‌ای دارد. برای مثال هر کدام از متدهای آپلود یاد شده یک چنین امضایی دارند:

```
[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 متد برای سه پارامتر یاد شده نیاز است)، ابتدا HttpPostedFileBase را در پوشه‌ایی که مدنظر دارید ذخیره کنید. سپس باید محتوایی را به RedActor بازگشت دهید و اصل کار یکپارچگی با ASP.NET MVC نیز در 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: ۱۳۹۱/۰۵/۰۴ ۳:۹

آقای نصیری واقعا شرمنده کردی استاد!

واقعا انتظار نداشتم که به این سرعت پاسخ بدید.
یک دنیا ممنونم

نویسنده: امیرحسین
تاریخ: ۱۳۹۱/۰۵/۰۴ ۳:۳۹

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: ۱۳۹۱/۰۵/۰۴ ۹:۵

سلام

آقای نصیری میشه طریقه استفاده از این ادیتور رو برای asp.net webform هم شرح بدهید؟

باتشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۰۴ ۱۰:۲۱

در وب فرم‌ها:
- یک TextBox را به صفحه اضافه کنید. TextMode آن باید MultiLine باشد تا تبدیل به TextArea شود. همچنین ClientID آن‌را هم مقدار دهی کنید تا بشود در jQuery ازش استفاده کرد.
- تمام توضیحات یکی است با این تفاوت که:
الف) return Content در اینجا می‌شود Response.Write
ب) بجای کنترلر شما از یک http handler می‌تونید 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 ghomi
تاریخ: ۱۳۹۱/۰۵/۰۴ ۱۹:۱۳

سلام من در Action مقدار زیرو برمی‌گردونم ولی در RedActor نشون نمیده

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: ۱۳۹۱/۰۵/۰۴ 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 مثلا 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 نباید 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

filename یک مسیر نسبی است (نسبت به ریشه سایت) مثلا:
GIAU/Images/Actors.png
یا مثلا (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** رو در سورس redactor (فایل **js**. redactor) پیدا نمی‌کنم. مطمئن هستید تداخلی در برنامه ندارید؟
ضمنا 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، کاملا تفاوت کرده و قسمت‌هایی از مطلب فوق دیگر معتبر نیست.
RedActor اینبار به خروجی json نیاز دارد. برای مثال به فایل مثال image_upload.php 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 array مثال PHP 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
لطفا این اصلاحات رو توی اصل متن اصلاح کنید
چون همه کامنت‌ها را تا انتها 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 از literal برای نمایش محتویات 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>aaaaaaa...").
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۱۶ ۱۷:۱۴

مراجعه کنید به قسمت 17 سری MVC سایت؛ قسمت «بررسی اعتبار درخواست (Request Validation) در ASP.NET MVC».

نویسنده: احمد احمدی

تاریخ: ۱۳۹۱/۰۶/۲۱ ۱۵:۳۸

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 پروژه‌ی شخصی ، بار دیگر هم در یک پروژه‌ی 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 باشید، 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 دکمه‌ی 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: ۱۳۹۱/۰۶/۲۷ ۱۴:۶

نیاز هست افزونه نویسی کنید. استاندارد نیست.

نویسنده: کیارش سلیمان زاده
تاریخ: ۱۳۹۱/۱۰/۲۰ ۱۲:۴۷

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: ۱۳۹۱/۱۰/۲۰ ۱۳:۱۱

آخرین نسخه‌ای که من دارم: redactor803.zip

نویسنده: کیارش سلیمان زاده
تاریخ: ۱۳۹۱/۱۰/۲۰ ۱۳:۱۵

ممنون وحید جان،
اینم هستش.

نویسنده: Mohsen
تاریخ: ۱۳۹۱/۱۱/۱۱ ۲۳:۲۹

با سلام
زمانی که از این ادیتور استفاده می‌کنیم آیا کاربر می‌تواند فونت را تغییر دهد؟ ممنونم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۱۱ ۲۳:۴۲

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 می‌تونید براش افزونه بنویسید تا اینکار را انجام دهد.

نویسنده: Mohsen
تاریخ: ۱۳۹۱/۱۱/۱۱ ۲۳:۵۲

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: ۱۳۹۱/۱۱/۱۲ ۰:۶

من چیزی رو از فایل css آن حذف نکردم. 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 css 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: ۱۳۹۱/۱۲/۲۰ ۱۹:۰

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: ۱۳۹۱/۱۲/۲۰ ۱۹:۹

روش تشخیص رو اینجا توضیح دادم. نیاز به json داره.

نویسنده: ناصر
تاریخ: ۱۳۹۱/۱۲/۲۰ ۱۹:۵۳

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 رو 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: ۱۳۹۱/۱۲/۲۰ ۲۱:۴۶

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

همچنین داخل این راهنما خروجی هم دقیقا آئرده شده که من بعد از سوال پیدا کردم. خروجی برای image باید به شکل زیر باشه

```
{ "filelink": "/images/img.jpg" }
```

و برای فایل هم به شکل زیر باشه

```
{ "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 وقتشون زیاد گرفته نشه :)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۲۱ ۲۲:۵۹

یک نکته: سعی نکنید این فرمت‌ها 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 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 لایسنس 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 رو میزنیم ، هیچ نسخه free وجود ندارد!
3 تا گزینه buy a licence وجود دارد و یک گزینه trial version
ممنون 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 کردم:

```
[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 نمی‌دهد.
ایراد کارم از کجاست؟ باتشکر.

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. newFileName 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 src تگ img.

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 کرده‌اند.

نویسنده: محمدرضا 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 paste_code.html آن در مسیر plugins:

```
<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 setColorNone

```
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 public.js و default.js ذیل دکمه justify

```
code:
{
    title: 'Code',
    func: 'showCodesPage'
},
```

به css 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: محمدرضا 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 می‌شه !
و متاسفانه از این مرحله بیشتر نتونستم 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 js نهایی، مشاهده کنید. کدهای ارسال شده با مقداری تغییر در نگارش‌های بعدی قابل استفاده هستند.

نویسنده: محمدرضا برنتی

تاریخ: ۱۳۹۲/۰۹/۱۵ ۱:۳۱

همین الان کارم تموم شد و تونستم دکمه code و با نگارش جدید این ادیتور هماهنگ کنم و در راه حل مشکلم تقریبا یاد گرفتم چجوری باید برای این ادیتور افزونه بنویسم ... و این بدون کمک شما امکان پذیر نبود ...
خیلی ممنونم ...

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۹/۱۵ ۱۰:۲۳

اگر دوست داشتید فایل نهایی خودتون رو پیوست کنید. ممنون.

نویسنده: پرویز
تاریخ: ۱۳۹۲/۰۹/۲۷ ۲۲:۲۷

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("../uploade/"),
Path.GetFileName(file.FileName));
                     file.SaveAs(filePath);
                }
              return Content("<img src='/uploade/" + file.FileName + "' />");
            }
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۲۷ ۲۲:۳۲

- 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: ۱۳۹۲/۱۲/۰۱ ۱۶:۵۰

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: ۱۳۹۲/۱۲/۰۱ ۱۶:۵۳

در نگارش جدیدتر آن این فایل‌ها با همان فایل اصلی ادغام و یکی شده‌اند.

نویسنده: فرزین بیاتی
تاریخ: ۱۳۹۲/۱۲/۰۱ ۱۷:۱۷

ممنون از پاسختون
من از ورژن v8.0.3 استفاده میکنم
میشه لطفاً بیشتر توضیح بدید که مثلاً چه جوری از فایل paste_code.html باید 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 آن‌را با کدهایی که کمی بالاتر ارسال کردم، تطابق داده و تکمیل کنید.

نویسنده: فرزین بیاتی
تاریخ: ۱۳۹۲/۱۲/۰۶ ۱۴:۳۷

دوستان فرق linkFileUpload و fileUpload چیه و با کدوم 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 مستقلا دکمه‌ی سنجاق مانندی در نوار ابزار دارد.

نویسنده: فرزین بیاتی
تاریخ: ۱۳۹۲/۱۲/۱۴ ۱۵:۳۳

چطوری میشه redactor رو readonly کرد؟ من readonly و disabled رو به TextArea 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: جوادنبی
تاریخ: ۱۳۹۲/۱۲/۲۵ ۱۸:۵۳

این کد من که نوشتمش 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/pmg" };
     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: ۱۳۹۲/۱۲/۲۶ ۲۲:۲۲

با سلام ، من می‌خواستم 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: جوادنبی
تاریخ: ۱۳۹۲/۱۲/۲۷ ۲۳:۱۰

با سلام؛ من یک ویو دارم که در آن از دوتا redactor استفاده کردم. در واقع سناریوی 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 partial اونredactor 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: ۱۳۹۲/۱۲/۲۷ ۲۳:۳۶

لینک تولید شده نهایی در اکشن متد را با لینک واقعی فایل بر روی سرور تطابق دهید.

نویسنده: پرویز
تاریخ: ۱۳۹۳/۱۰/۱۶ ۱۲:۳۴

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.
سوال: چطوری میشه با حذف عکس از درون ویرایشگر خود عکس هم از روی هاست پاک بشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۱۶ ۱۲:۳۸

چنین قابلیتی را ندارد. برای این مورد از Froala WYSIWYG Editor استفاده کنید (^).

نویسنده: محمد محمدی
تاریخ: ۱۳۹۳/۱۱/۲۶ ۲۳:۳۰

در وب فرم با چیزی که گفتید پیش رفتم و ذخیره فایل به درستی انجام میشه؛ ولی هیچ چیزی برگردانده نمیشه. آیا نیاز به تنظیمات خاصی داره؟ به این خاطر میگم چون مقدار imageUpload در فایل redactor.js برابر false هستش.

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۱۱/۲۷ ۰:۴

اگر نظرات رو مطالعه کنید، چندین نگارش از این افزونه موجود هست که خروجی‌های متفاوتی رو نیاز دارند. ضمنا بهتره این افزونه رو با یک نمونه‌ای که مدام داره به روز میشه تعویض کنید.

عنوان: معرفی پروژه Orchard
نویسنده: حسام امامی
تاریخ: ۱۳۹۱/۰۵/۰۶ ۱۳:۱۶
آدرس: www.dotnettips.info
گروه‌ها: MVC, Orchard, CMS

معرفی پروژه Orchard:

سیستم مدیریت محتوای Orchard توسط مایکروسافت در ژانویه سال 2011 همراه با ASP.NET MVC 3, IIS Express, SQL CE 4 ,فریم ورک Web Farm و WebMatrix ارائه شد. هدف تمامی این پروژه‌ها ایجاد قابلیتی برای توسعه آسان برنامه‌های تحت وب در محیط ویندوز بود. 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 CodePlex Foundation نیز شناخته می‌شود) است که توسط مایکروسافت پشتیبانی می‌شود. Outercurve Foundation یک سازمان غیر انتفاعی است که هدف آن تشویق و حمایت از پروژه‌های متنی بازی نظیر Orchad و یا toolkit معروف ASP.NET MVC یعنی MVC Contrib است. مایکروسافت به صورت رسمی از Orchad پشتیبانی نمی‌کند اما در حال حاضر برنامه نویسانی را جهت توسعه این سیستم استخدام کرده است.

برای پروژه Orchad 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 مرجع با استفاده از کامپوننت‌های فوق
3)ساخت انجمن هایی برای پشتیبانی از این کامپوننت‌ها و یا برنامه‌های مرجع

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 متن باز نوشته شده است. Orchard از ASP.NET MVC 3.0 به همراه View engine جدید و فوق العاده آن یعنی Razor بهره می‌برد. همچنین این پروژه وابستگی زیادی به دیگر ابزارهای متن 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 ماژول و تم برای آن ساخته شده است که در گالری آن در دسترس می‌باشد. Orchard به صورت ریلیزهای جزئی ارائه می‌شود و جدیدترن نسخه آن در هنگام نوشتن این متن 1.5.1 می‌باشد.

ا ما چرا به یک CMS دات نتی دیگر نیاز است ؟

تعداد زیادی سیستم‌های مدیریت محتوای تجاری و یا متن 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 متن 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 نتی به سمت ASP.NET MVC مهاجرت کرده اند به همین دلیل سیستم Orchard بر مبنای این تکنولوژی نسبتا جدید دات نت پیاده شده است. با استفاده از Orchard می‌توان یک وب سایت با عملکرد بسیار بالا بدون نوشتن حتی یک خط کد ایجاد نمود. اما مانند هر سیستم مدیریت محتوی دیگری اگر بخواهیم به آن قابلیت هایی 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: ۱۳۹۱/۰۵/۰۷ ۷:۵۸

سپاس

در صورت امکان مقایسه ای بین این پروژه و kooboo که آن هم پروژه cms با asp.net mvc3 هست داشته باشید

نویسنده: حسام امامی
تاریخ: ۱۳۹۱/۰۵/۰۷ ۱۲:۴۹

kooboo حدود 5 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 بتونند کمک کنند

نویسنده: دانشور
تاریخ: ۱۳۹۱/۰۵/۱۷ ۹:۱

خواهشمندم در صورت امکان راجع به معماری پروژه فوق مطلب بنویسید .

نویسنده: حسام امامی
تاریخ: ۱۳۹۱/۰۵/۱۷ ۱۲:۵۹

با سلام
اتفاقا 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 حتما بعد از پایان آموزش Git (شاید حدود 3 هفته دیگه) تمام جزئیات Orchard به همراه نحوه ساخت ماژول و تم رو آموزش می‌دهم

نویسنده: دانشور
تاریخ: ۱۳۹۱/۰۵/۱۸ ۱۱:۲۳

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: ۱۳۹۱/۰۵/۱۸ ۱۱:۲۷

این پروژه سورس باز است؛ نیازی به مهندسی معکوس ندارد.

نویسنده: سازگار
تاریخ: ۱۳۹۱/۰۵/۲۱ ۲۱:۵۱

ممکن است چند سایت را که توسط orchard ساخته شده است اعلام کنید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۲۱ ۲۳:۵۰

Orchard Top Sites

Sites on Orchard

نویسنده: سازگار
تاریخ: ۱۳۹۱/۰۵/۲۵ ۱۱:۵۵

از لطف شما ممنون هستم. سایتهای معرفی را چک کردم بدون استثنا 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 (4 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 داشته‌باشند 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 بفهمم در زمان اجرا چه اتفاقی می‌افتد . dashboard آن چگونه فراخوانی میشود و چگونه می‌توان رابط کاربری dashboard را مطابق میل خود طراحی مجدد کرد. در این زمینه میتوانید مرا کمک کنید و یا منابعی را که راجع به سورس کد آن توزیع میدهد معرفی کنید ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۲۵ ۱۹:۴۷

پشنیاز درک کدهای آن مطالعه قسمت‌های MVC و NHibernate سایت است (به علاوه مباحث تزریق وابستگی‌ها). هر کدام یک 30 قسمتی هستند. مطالعه و تمرین آن‌ها حدودا 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 ازتوجه شما سپاسگزارم

نویسنده: حسام امامی
تاریخ: ۱۳۹۱/۰۵/۲۵ ۲۰:۳۰

یافتن مستندات درباره طرز کار یک CMS در یک جا عملا غیر ممکن است حتی CMSهای قدیمی‌تر نیز چنین داکیومنت هایی 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 قطعا orchard شروع خوبی نیست

نویسنده: mahdi1391
تاریخ: ۱۳۹۱/۰۸/۱۶ ۱۹:۴۱

برای نمونه سایت زیر را مشاهده کنید، سرعت بسیار بالایی دارد http://www.bradygaster.com

نویسنده: حمید
تاریخ: ۱۳۹۲/۱۰/۰۶ ۴:۳۵

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 لایسنس هم که عالی هست حتی میشه Badge of Honor رو هم تغییر داد.

عنوان: مباحث تکمیلی مدل‌های خود ارجاع دهنده در EF Code first
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۳ ۱۴:۲۴
آدرس: www.dotnettips.info
گروه‌ها: Entity framework, MVC

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 باشند. 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 چنین جدولی با اطلاعاتی که ملاحظه می‌کنید تشکیل خواهند شد:

یک نظر ارائه شده و سپس دو نظر تو در توی دیگر برای این نظر ثبت شده است.

اولین نکته اضافه‌تری که نسبت به قسمت هشتم قابل ملاحظه است، تعریف خاصیت جدید 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 متد RunTests دقت کنید، ابتدا یک ToList نوشته شده است. این مورد سبب می‌شود که تمام رکوردهای مرتبط دریافت شوند. مثلا 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 EF هماهنگ شود (متد 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: ۱۳۹۱/۰۵/۲۶ ۹:۲۷

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 متد Seed یا معرفی اولین شخص با توجه به این که این RelatedPerson نمیتواند خالی باشد چطور میتوان خود شخص را به این پراپرتی معرفی کرد و بعنوان روت اشخاص از آن استفاده کرد و مپ آن به چه شکل است.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۲۶ ۹:۳۱

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: ۱۳۹۱/۰۵/۲۶ ۹:۳۵

در مورد NHibernate هم این محدودیت وجود دارد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۲۶ ۹:۳۸

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: ۱۳۹۱/۰۵/۲۶ ۹:۵۴

در مورد تعریف ORM حق با شماست ولیبا 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 مثلا شناسه اون حتما با یک ذخیره میشه.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۲۶ ۹:۵۷

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 کلیدخارجی 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 فیلد ReplyId است که نال پذیر است. این ReplyId کلید خارجی اشاره‌کننده به همین جدول هم هست.

نویسنده: مهدی پایروند

تاریخ: ۱۳۹۱/۰۵/۲۶ ۱۰:۶

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]
GO
```

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 parent-id 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: ۱۳۹۱/۰۵/۲۶ ۱۰:۱۷

parent-id کلید خارجی است. برای مقدار دهی آن (ثبت اولیه رکورد مرتبط) اگر null پذیر نباشد نیاز است حتما رکورد اشاره کننده به آن وجود خارجی داشته باشد.
به همین دلیل باید در این حالت خاص آن‌را نال پذیر تعریف 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: ۱۳۹۱/۰۵/۲۶ ۱۰:۳۷

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. ضمنا امکان مقدار دهی دستی ParentId هم در اینجا تا زمانیکه رکورد ثبت نشده باشد، میسر نیست (کاری که management studio به صورت دستی انجام داده، چند مرحله کار بوده نه صرفا یک insert معمولی).

نویسنده: مهدی پایروند
تاریخ: ۱۳۹۱/۰۵/۲۶ ۱۰:۴۵

بله ممنون، همین موردی که میگید چند مرحله رو باید بشه با ORM بدست آورد، با SQL Profiler میشه از کارکرد SQL Managment نمونه برداری کرد، البته شاید این کار خروجی مناسبی برای سمت کد نداشته باشه.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۲۶ ۱۰:۵۰

طراحی مدنظر شما اصلا 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: ۱۳۹۱/۰۹/۱۷ ۲۳:۴۳

با سلام؛
اگر ما در مدل یک فیلد به عنوان مثال status داشته بایم و بخواهیم آنهایی که status شان 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 باوضعیت ok داشته باشه فرزندانش با هر وضعیتی باش میاره
راه حلی به نظرتون میاد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۹/۱۸ ۰:۱۴

همان ابتدای کار از && برای فیلتر کردن استفاده کنید

```
p => p.PostId == postId && p.ComentStatus == ComentStatus.Ok
```

نویسنده: میلاد محسنی
تاریخ: ۱۳۹۱/۱۱/۱۹ ۲۲:۲۹

با سلام.
برای تهیه مدل‌های خود ارجاع دهنده، نظرتان در خصوص استفاده از hierarchyID چیست؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۱۹ ۲۳:۶

قابل انتقال نیست به سایر بانک‌های اطلاعاتی. یکی از اهداف کار با ORMها مستقل کردن برنامه از بانک اطلاعاتی است. مثلا 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: ۱۳۹۱/۱۲/۰۱ ۱۹:۴۴

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 لاگ 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 میشه 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 نتونستم parent را برای viewmodel به خوبی تعریف کنم.
ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۰۱ ۲۱:۲۹

- این خروجی SQL لاگ 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 در ToList اول، کار با دیتابیس تمام و قطع می‌شود. ToList 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 پیدا کردید ... حتما مشکلی در تنظیمات نگاشت‌ها دارید.
- اگر duplicate reader دارید شاید بخاطر lazy loading سایر خواص راهبری است که تعریف کردید مانند User و EditByUser و غیره. این‌ها اگر قرار است نمایش داده شوند، پیش از ToList اول باید توسط متد الحاقی Include به صورت eager loading تعریف شوند تا lazy loading و duplicate reader نداشته باشید.
- برای فیلتر فیلدهای اضافی، پیش از ToList اول، با استفاده از Projection و نوشتن یک Select، موارد مورد نیاز را انتخاب کنید.

نویسنده: مهدی سعیدی فر
تاریخ: ۱۳۹۱/۱۲/۰۱ ۲۲:۳۰

ممنون آقای نصیری

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 می‌تونید 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.

ممنون

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 می‌تونه به صورت غیر بازگشتی بگه ممنونش میشم.
این کد یه partialpage برای navbar هست که هرکسی برای bootstrap به راحتی میتونه 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></text>
                @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 فیلدهای اضافی هم کوئری گرفته میشه.میدونم فیلدهای 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؟

ممنون

نویسنده: ali.rezayee
تاریخ: ۱۳۹۲/۰۱/۱۴ ۱۳:۵۶

با سلام و تشکر از 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 رابطه‌ها شرکت نکنند، یعنی فقط تعدادی فیلد از جدول BlogComment انتخاب شود. در حالت فعلی تمام فیلدها از تمام جدولهای مرتبط استخراج میشوند.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۱/۱۴ ۱۴:۴۰

- جدول نظرات اگر با موجودیتی به نام کاربر سر و کار دارد، این خاصیت عموما به صورت virtual تعریف می‌شود یعنی lazy loading قرار است رخ دهد. به عبارتی با ToList اولیه اگر از متدی مانند 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 دیباگر VS.NET اگر نود مربوط به User رو باز کنم، اطلاعاتش هست. پاسخ این است که بله. دقیقا در همین لحظه که نود رو باز کردید یک کوئری برای دریافت اطلاعات یوزر به سرور ارسال شده و نه پیش از آن. البته می‌شود lazy loading را در EF کلا خاموش کرد یا حتی به صورت مقطعی با استفاده از متدی مانند AsNoTracking. اما حالت پیش فرض دقیقا چیزی است که عنوان شد.

بنابراین درحالت فعلی تمام فیلدهای وابسته از بانک اطلاعاتی استخراج نمی‌شوند مگر اینکه lazy loading را به نحوی تبدیل به eager loading کرده باشید.
- ضمنا در حالت فعلی اگر دقت کرده باشید پیش از ToList اول یک سه نقطه گذاشته شده است. یعنی اینجا می‌تونید where بنویسید. می‌تونید Select بنویسید و به صورت اختصاصی یک سری خاصیت مشخص رو انتخاب کنید و خیلی از کارهای دیگر.

نویسنده: ali.rezayee
تاریخ: ۱۳۹۲/۰۱/۱۴ ۱۵:۱۹

با تشکر فراوان از پاسخ کامل جنابعالی.
مشکل من از اینجا شروع شد که با Json.Net 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.
ممنون 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: ۱۳۹۲/۰۱/۱۴ ۱۶:۱۰

از ویژگی Newtonsoft.Json.JsonIgnore برای عدم serialization یک خاصیت خاص به JSON استفاده کنید.

نویسنده: ali.rezayee
تاریخ: ۱۳۹۲/۰۱/۱۴ ۲۱:۴۰

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 JsonNetResult Index()
        {
            var ctx = new testEntities();
            var list = ctx.BlogComments
                .Where(p => p.Id == 1)
                .Select(p => new
                                {
                                    p.Id,
                                    p.Body,
                                    p.Children
                                })
            .ToList();

            JsonNetResult jsonNetResult = new JsonNetResult();
            jsonNetResult.Formatting = Formatting.Indented;

            jsonNetResult.SerializerSettings = new JsonSerializerSettings()
                                                    {
                                                        ReferenceLoopHandling =
ReferenceLoopHandling.Ignore
                                                    };
            jsonNetResult.Data = list;
            return jsonNetResult;
        }
```

```
[
  - {
      Id: 1,
      Body: "1 نظر",
    - Children: [
        - {
            - Children: [
                - {
                    Children: [ ],
                    Id: 4,
                    Body: "1-2 نظر",
                    DateSend: "2013-03-03T00:00:00",
                    DateRead: null,
                    IsDeleted: false,
                    UserId: 1,
                    ReplyId: 2
                  }
              ],
              Id: 2,
              Body: "1-1 نظر",
              DateSend: "2013-03-03T00:00:00",
              DateRead: null,
              IsDeleted: false,
              UserId: 1,
              ReplyId: 1
            }
      ]
    }
]
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۱/۱۴ ۲۲:۵۹

علت این است که p.Children به تمام خواص عمومی شیء BlogComment اشاره می‌کند؛ این رو هم میشه یک سطح دیگر با Projection سبک‌تر کرد و یا بجای Projection در حالت شما ساده‌تر است که JsonIgnore را روی تمام خواصی قرار دهید که نباید توسط JSON.NET بررسی شود. با توجه به lazy loading، این خواص virtual توسط EF در بدو امر بارگذاری نمی‌شوند و

همچنین چون توسط JSON.NET به دلیل JsonIgnore معرفی شدن واکاوی مجدد نخواهند شد، بنابراین مشکلی از لحاظ کارآیی یا حجم بالای خروجی نخواهید داشت.

نویسنده: ali.rezayee
تاریخ: ۱۳۹۲/۰۱/۱۴ ۲۳:۳۲

ممنون از لطف شما.
عالی و جامع بود.

نویسنده: مولانا
تاریخ: ۱۳۹۲/۰۲/۱۲ ۲۰:۳۹

دوست عزیز.
از مطلب مفید شما متشکرم. من هم مشکل شما رو دارم. آیا به جواب رسیدید؟

نویسنده: مولانا
تاریخ: ۱۳۹۲/۰۲/۲۴ ۲۲:۴۶

باسلام.
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؟ (می خواهم با حذف یک فولدر تمام محتویات آن نیز حذف شوند.)

نویسنده: ایمان محمدی
تاریخ: ۱۳۹۲/۰۴/۱۰ ۱۱:۳۱

سلام
در مثال بالا چون نظرات با یک پست مرتبط هستند قاعدتا براساس یک پست خاص کامنت‌ها رو 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 sql server محدود می‌کنه و 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
    WHEREID = @ID
    UNION ALL
    SELECTt.*
    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 (مثلا 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; //عدد 4 در اینجا شماره رکورد کامنتی است که دارای یک سری زیر کامنت است
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

ممنون 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
person4 {name="hamid",GroupId=4}//فروش قطعات
person5 {name="hasan",GroupId=4}//فروش قطعات
person6 {name="Ahmad",GroupId=5}//خدمات
person7 {name="vahid",GroupId=1}//گروه اصلی
```

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 idها 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

تاریخ: ۱۳۹۲/۰۴/۱۹ ۲۱:۲۱

برای تکمیل بحث، روش مایکروسافت برای حل مساله مدل‌های خود ارجاع دهنده:
Recursive or hierarchical queries using EF Code First and Migrations
از نوشتن مستقیم SQL و ایجاد View استفاده کرده.

نویسنده: ali.rezayee
تاریخ: ۱۳۹۲/۰۴/۲۴ 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 تا 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 واکشی و مقدار امتیاز آن به روز شوند تا به ریشه برسیم؛ راه حل دیگر نوشتن تریگر در سمت دیتابیس است. فکر میکنم در هر دوحالت سربار زیادی داشته باشیم .آیا راه بهتری وجود دارد؟
ممنون

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۴/۲۴ ۱۴:۷

کم هزینه‌ترین روش: جمع‌ها رو سمت کلاینت مدیریت کنید و اصلا 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: ۱۳۹۲/۰۴/۲۴ 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: ۱۳۹۲/۰۴/۲۴ ۱۴:۳۲

برای نمایش اول کار: جمع کل ریشه اصلی مساوی است با جمع فرزندانی که والد غیر null دارند. یک کوئری سبک بیشتر نیست. مابقی جمع‌های درخت رو میشه سمت کلاینت مدیریت کرد.

نویسنده: امیرحسین جلوداری
تاریخ: ۱۳۹۲/۰۵/۰۷ ۱۳:۲۲

روش بهینه‌ی حذف پدر و فرزندان اون چیه؟! ... ممنون میشم اگه رو همین مثال لطف کنین

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۵/۰۷ ۱۳:۳۰

- نمی‌تونید والد رو یک ضرب حذف کنید. از آخرین فرزند به اول ریشه (والد ریشه جاری) باید (یکی یکی) حذف کنید تا قیود تعریف 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: ۱۳۹۲/۰۵/۰۷ ۱۳:۳۲

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: ۱۳۹۲/۰۵/۰۷ ۱۴:۲۳

روش شما رو تست کردم ولی جواب نداد!... فکر میکنم این ساختاری که شما تعریف کردید برای حالت cascadeDelete جواب میده ...

طبق فرمایش آقای نصیری کاری که من کردم اینه (روی سیستم گروه و زیرگروه) :

1 - واکشی 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: ۱۳۹۲/۰۵/۰۷ ۱۵:۱۶

نه کار میده. اون هم در حالتی که cascade نباشه.

این کدی که نوشتم دقیقا مال پروژه‌ی IRIS هست. یک بار کدش را در اون پروژه ببینید؛ ضرری نداره.

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۳/۰۳/۱۷ ۱۱:۱۸

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;
```

نویسنده: پریسا زاهدی
تاریخ: ۱۳۹۳/۱۲/۰۳ ۲۰:۳۹

سلام

- شما برای نمایش نظرات چند سطحی سایت جاری چه روشی را بکار برده اید و پیشنهاد می‌کنید (در پروژه‌های MVC و EF CF) ؟

- استفاده از Recursive CTE برای نمایش نظرات چند سطحی روش مناسبیه ؟

ممنونم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۲/۰۳ ۲۰:۴۴

- مطلب جاری دقیقا خلاصه‌ی کاری است که انجام شده؛ به همراه روش نمایشی آن که در انتهای بحث ذکر شد.

- بله. البته اگر بخواهید مستقیما SQL بنویسید، دیگر نیازی به ORMها نخواهد بود و خیلی از قابلیت‌های بومی دیتابیس‌ها امکان انتقال ندارند و پروژه را وابسته به یک دیتابیس خاص می‌کنند.

عنوان: **سازگار کردن لینک‌های قدیمی یک سایت با ساختار جدید آن در ASP.NET MVC**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۱۴ ۲۱:۲۴
آدرس: www.dotnettips.info
برچسب‌ها: ASP.Net, MVC, ASP.Net MVC

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 بوده، برای نمونه گوگل هر از چندگاهی مجددا به این آدرس مراجعه می‌کند تا حداقل مطمئن شود وجود خارجی دارد یا خیر (این نکته را از لاگ‌های خطای سایت استخراج کردم).
- سایت‌های زیادی هستند که پیشتر به سایت شما و مطالب آن لینک داده‌اند. نمی‌توانید از آن‌ها درخواست کنید لطفا 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 آن‌را جهت سازگاری سایت جاری با هاست قبلی آن (بلاگر) در ادامه مطالعه خواهید نمود:

الف) سازگار سازی لینک‌های قدیمی برچسب‌های سایت با ساختار جدید آن

در بلاگر 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 متدی به نام Index در کنترلر Tag که دارای پارامتری به نام name است ارسال خواهد شد.

ب) از دست ندادن خوانندگان قدیمی فیدهای سایت

دو نوع فید کلی در بلاگر وجود دارد: http://site/feeds/posts/default و http://site/feeds/comments/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 درنظر 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 کلا 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، مجددا بدون مشکل قابل استفاده بوده و همچنین موتورهای جستجو نیز امکان به روز رسانی اطلاعات خود را خواهند یافت.

عنوان: CAPTCHAfa
نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۰۵/۱۵ ۱۲:۴۶
آدرس: www.dotnettips.info
گروه‌ها: CAPTCHA, reCAPTCHA, CAPTCHAfa, ASP.Net, MVC, ASP.Net MVC

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 متون 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 انستیتو تکنولوژی ایلینویز شیکاگو، پروژه ای بر همین اساس اما برای متون فارسی با عنوان CAPTCHAfa تولید کرده اند (شکل زیر) که در این آدرس در دسترس است. امیدوارم این پروژه به گونه ای تغییر کنه که برای دیجیتالی کردن متن‌های پارسی استفاده بشه. در حال حاضر، این پروژه از کلماتی از پیش تعریف 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.

```
// =-=-=-=
```

```
            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 روال Page_Load، کد زیر را قرار دهید. این کار برای تزریق اسکریپ CAPTCHAfa به صفحه استفاده می‌شود.

```
if (!IsPostBack)
{
    litCaptcha.Text = Captchafa.CaptchafaGetHtml();
}
```

litCaptcha، یک کنترل Literal است که اسکریپت تولید شده، به عنوان متن آن معرفی می‌شود.

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
}
else
{
   // پاسخ صحیح نیست
```

```
}
```

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.Captchafa();

<input type="submit" id="btnCaptchafa" name="btnCaptchafa" 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 recaptcha 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 "مسل 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 سایت‌های فارسی مرسوم بشه

نویسنده: علیرضا اسم‌رام
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 جزییات را دقیق ببینم، اما امیدوارم تلفظ و پخش کلمات بصورت صوتی خوب عمل کند!

نکته دیگر اینکه در مورد پیچیدگی تصویر CAPTCHA، بهترین روش استفاده از نقطه و پاره خط جهت تشویش تصویر است. معمولاً

بیشترین خطاها 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 بیضی هایی 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 کردن 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 شیوه ای 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 دیجتالی 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 CAPTCHAیی مانند reCAPTCHA که بر اساس دیجیتالی کردن متون پیاده سازی شدند، دو کلمه رو به کاربر نشان میدن:
1) کلمه ای که توسط رایانه به درستی تشخیص داده شده
2) کلمه یا بخشی از اون که رایانه نتونسته اون رو به درستی تشخیص بده.
در صورتی که کاربر، کلمه ای 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، متد CaptchafaGetHtml به GetHtml تغییر پیدا کرد، مدیریت خطاها 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

نویسنده: محمد رضا صفری

تاریخ: ۱۳۹۳/۱۱/۲۰ ۱۳:۱۲

فکر کنم که کلا پروژه تعطیل شده ...

عنوان: **نکته‌ای در استفاده از AutoMapper**
نویسنده: سیدمجتبی حسینی
تاریخ: ۱۳۹۱/۰۵/۲۱ ۲:۵۳
آدرس: www.dotnettips.info
گروه‌ها: MVC, AutoMapper

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: ۱۳۹۱/۰۵/۲۱ ۹:۲۶

سلام ،

در AutoMapper معرفی نگاشت‌ها از پیش شرط‌های انجام نگاشت اشیا به هم می‌باشد.
معمولا این نگاشت‌ها در ابتدای شروع برنامه (مثلا متد Application_Start در Global.asax) تعریف می‌شوند.

نویسنده: مجتبی حسینی
تاریخ: ۱۳۹۱/۰۵/۲۱ ۱۱:۵۷

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: ۱۳۹۱/۰۵/۲۱ ۱۲:۱۰

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 متد Application_Start هست."

نویسنده: محسن
تاریخ: ۱۳۹۲/۱۰/۲۷ ۱۷:۱

با سلام به دوستان عزیزم
نحوه انتقال اتوماتیک validation attributeهای هر مدل به ویومدل مربوطه چگونه توسط automapper قابل انجامه؟ چون به صورت پیش فرض data annotationها مپ نمیشن، آیا تنظیم خاصی برای این کار مورد نیازه؟

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۲۷ ۱۷:۲۵

قبل از پرسش بهتره کمی جستجو کنید. مثلا یک مورد در اینجا بحث شده

نویسنده: ح مراداف
تاریخ: ۱۳۹۳/۱۱/۰۷ ۱۷:۷

سلام؛ در تمامی مثالها شما از مپر درون اکشن‌ها استفاده کردین، مگر لایه سرویس نباید خودش یک ویو-مدل بگیره و بقیه کارهای نگاشت و ... رو انجام بده و در خروجی نیز نوع ویو-مدل خروجی دهد ؟
من یکم گیج شدم. احساس می‌کنم عملیات اتومپر باید درون لایه سرویس انجام بشه که داخل اکشن فقط متدهای سرویس رو صدا بزنیم. از طرفی هم توی لایه سرویس که اومدم استفاده کنم که ویو-مدلم رو به کلاس ef نگاشت کنم و بعد بریزم توی دیتابیس دیدم که نمیشه ، چون لایه سرویس application_start برای کانفیگ اتومپر نداره ؟! لطفا راهنمایی نمایید. باتشکر

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۱۱/۰۷ ۱۸:۱۹

لایه سرویس با بقیه قسمت‌های برنامه درون یک app domain اجرا میشن. یعنی همینکه تنظیمات اولیه auto mapper رو در ابتدای برنامه انجام دادید، بقیه لایه‌ها هم می‌تونن ازش استفاده کنن. این تنظیمات singleton هستن. یعنی فقط یک وهله ازشون در کل

طول عمر برنامه وجود داره.

عنوان: **یکپارچه سازی CKEditor با Lightbox**
نویسنده: مهدی سعیدی فر
تاریخ: ۱۳۹۱/۰۵/۲۸ ۲:۸
آدرس: www.dotnettips.info
برچسب‌ها: MVC, CKEditor

در یک پروژه من احتیاج داشتم تا عکس هایی که کاربر از طریق ckeditor آپلود می‌کنه ، به صورت خودکار با lightbox یکپارچه شه و به صورت گالری عکس نمایش داده شود.
همان طور که می‌دونید مکانیزم عملکرد اکثر پلاگین‌های lightbox به صورت زیر است:

```
<a href="orginal size of image directory"><img src="thumb(smaller size) image directory" /></a>
```

خب معلوم میشه که آدرس عکسی که به صورت کوچیک (اصطلاحا بند انگشتی) به نمایش در میاد باید داخل تگ img ، وآدرس عکس با سایز اصلی ، داخل تگ a قرار بگیره.

پس تقریبا 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 کند، احتیاج داریم که آدرس عکس با سایز اصلی را داخل تگ a قرار دهیم، که در ckeditor می‌توانیم 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 عکس‌های آپلود شده به ckeditor و روش‌های 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 کاملا هم بدون اشکال نیستند ولی نیازهای شما را برآورده می‌کند.
ج)آدرس دادن فایل‌های من کاملا 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 فولدری مثلا به نام Scripts در پروژه‌ی خود بریزید.
همان طور که می‌دانید ckeditor با یک 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: حتما 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 میاد 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 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 مستقیما 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 ، چه پارامترهایی را به اکشن متد من 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 متدی که من برایش مشخص کردم را فهمیده.
اما سه پارامتر دیگر نیز به اکشن متد ما ارسال می‌کنه: CKEditor و CKeditorFuncNum و langCod
برای اینکه با این پارامترهای ارسالی بیشتر آشنا شوید ، توصیه می‌کنم این صفحه را ببینید.
آنچه که از این پارامترها برای ما مهم هست ، من در اکشن متد تعریفی خود لحاظ کرده ام.

خب همون طور که پیش از این گفته بودم ما نیاز داریم که عکس آپلود شده را به ابعاد کوچکتر تغییر اندازه داده و اصطلاحا 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 را برای کار کردن با عکس برایتان معرفی می‌کنم، به نام WebImage که در داخل فضای نامی System.Web.Helpers 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 متد UploadImage را برایتان قرار می‌دهم که زیاد هم بهینه 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 متد به 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 متد 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 برگردوندیم:

اولیش:

```
window.parent.CKEDITOR.tools.callFunction(" + CKEditorFuncNum + ", \"" + thumbPath + "\", \"" + message
+ "\");
```

در حقیقت ما در اینجا ما از apiهای 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 شایدم 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.getElementsById("cke_145_textInput").value = defaultPath;
```

اما بازم js شروع کرد به بدقلق شدن. هرچی توی کنسول دیباگش کردم خطای null بودن را میداد. بعد از یه ساعت سرو کله زدن و تقریبا ناامید 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.getElementsById("cke_145_textInput").value = defaultPath;
```

موفقیت در این قسمت از کد باعث شد که من دست به کدتر شوم و مشکل border عکس ها، که به صورت دیفالت در IE یا همون دشمن همیشگی وجود داره ، را حل کنم ومقدار border را به صورت پیش فرض صفر کنم.

```
window.parent.document.getElementsById('cke_125_textInput').value=0;
```

خب همه‌ی این دردسرها 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 ی که متن و عکس‌های ما را شامل میشه ، کلاسش 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

نویسنده: علی
تاریخ: ۱۳۹۱/۰۵/۲۸ ۱۸:۱۶

با سلام و تشکر ببخشید می‌خواستم دلیل شما را برای استفاده نکردن از Ckfinder را بپرسم

نویسنده: بهزاد شیرانی
تاریخ: ۱۳۹۱/۰۹/۲۲ ۲۰:۲

ckfinder مجانی نیست و امکانات زیادی به شما نمیده

نویسنده: 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 مثال(سورس پروژه) رو بذارید؟

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۴/۲۵ ۱۱:۵۶

پروژه IRIS از CKEditor هم استفاده کرده. یک نگاهی به سورسش داشته باشید.

نویسنده: مهدی سعیدی فر
تاریخ: ۱۳۹۲/۰۴/۲۵ ۱۵:۴

سلام
با فایرفاکس تست کنید تا اگه مشکلی داشتید راهنمایی کنم.

عنوان: **تهیه خروجی RSS در برنامه‌های ASP.NET MVC**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۲۰ ۱۷:۵۱
آدرس: www.dotnettips.info
گروه‌ها: MVC, RSS, ASP.Net MVC

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 rtl 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 دات نت است.
6) تنظیم صحیح ContentEncoding و ContentType جهت نمایش بدون مشکل متون 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 آن‌را 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. FeedItem هم یک چنین ساختاری دارد و در اسمبلی MvcRssHelper 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 مثلا 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) نهایتا 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 = "مطلب " + 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 متد mapPostsToFeedItems ملاحظه می‌کنید.

نکته مهم بکارگرفته شده در متد mapPostsToFeedItems، استفاده از Url.Action برای تولید آدرس‌هایی مطلق متناظر 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. ضمنا جهت کاهش بار سرور می‌توان از OutputCache نیز به مدتی مشخص استفاده کرد.

نظرات خوانندگان

نویسنده: علیرضا اسم‌رام
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 description به‌جای content استفاده کنند.آیا این کتابخانه این قابلیت رو داره؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۲۱ ۰:۴۳

خروجی کتابخانه فوق، RSS استاندارد است و description دارد. مقادیر خواص کلاس FeedItem نهایتا به این خروجی نگاشت می‌شود.
برای نمونه این خروجی رو می‌تونید در سورس نهایی فید سایت مشاهده کنید.

نویسنده: محمد صافدل
تاریخ: ۱۳۹۱/۰۶/۲۱ ۱۰:۳۰

ممنون 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
ممنوم بابت مطلب خوبتون
ولی فایل که برای دانلود گذاشتید مشکل داره
پروژه ام وی سی توش 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

پوشه MvcRssHelper : کتابخانه اصلی

نویسنده: مهدی پایروند
تاریخ: ۱۳۹۱/۰۷/۲۳ ۱۵:۲۲

سلام، من هم توی لود پروژه مشکل دارم، البته همه نسخه‌های mvc رو 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 ایی که تولید می‌شد با atom جایگزین شد.

نویسنده: امیرحسین مرجانی
تاریخ: ۱۳۹۱/۰۸/۱۳ ۱۷:۳

سلام

از این روش می‌شه برای ساخت sitemap هم استفاده کرد؟
اگر نه چطور؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۴/۰۱/۱۰ ۱:۵۵

یک نکته‌ی تکمیلی

فید سایت امروز از کار افتاده بود. علت آن وجود یک سری کاراکتر غیرمجاز XML در متن بود که باید به نحو ذیل پاک شوند:

```
private static readonly Regex _matchHexadecimalSymbols =
            new Regex("[\x00-\x08\x0B\x0C\x0E-\x1F]", RegexOptions.IgnoreCase | RegexOptions.Compiled);

        /// <summary>
        /// there are a lot of symbols which can't be in xml code.
        /// </summary>
        public static string RemoveHexadecimalSymbols(this string txt)
        {
            return string.IsNullOrWhiteSpace(txt) ?
                string.Empty : _matchHexadecimalSymbols.Replace(txt, string.Empty);
        }
```

در کدهای مطلب فوق، مقادیر content و title باید پیش از استفاده، توسط این متد پاکسازی شوند.

نویسنده: سانی
تاریخ: ۱۳۹۴/۰۱/۳۰ ۲۳:۲۱

خیلی متشکر، کدها بسیار کار امد بود و بدون دردسر در پروژه جواب داد
فقط این که مثلا باید ادرس http://localhost:40252/feed/rss را وارد کرد ، حال این شکل rss (مربع نارنجی قرمز بالا چپ ادرس بار در گوگل کروم)که در نوار ابزار نمایش داده میشود رو چه تغییراتی باید انجام بدهم تا در ادرس بار نمایش داده شود ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۴/۰۱/۳۱ ۰:۲۷

به هدر صفحه باید این مورد جهت فعال سازی auto discovery اضافه شود:

```
<link title="فید آخرین تغییرات سایت" href="/rss.xml" type="application/rss+xml" rel="alternate" />
```

محدودیتی هم ندارد. هر تعدادی که نیاز است.

عنوان: **ایجاد Helper سفارشی جهت نمایش ویدئو در ASP.NET MVC**
نویسنده: علیرضا اسم‌رام
تاریخ: ۱۳۹۱/۰۶/۲۲ ۰:۵۳
آدرس: www.dotnettips.info
گروه‌ها: MVC, ASP.Net MVC, Helper

معرفی HTML Helpers

در صورتی که در مورد Helperها در ASP.NET MVC 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 بخش هایی از آن را Optional کنیم تا در زمان استفاده از آن، نسبت به مقادیر این دسته از پارامترها تصمیم گیری 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 "دریافت 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 حرقی 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.ConvertColorToHexaString(youtubePlayerOption.PrimaryColor),
ConvertColorToHexa.ConvertColorToHexaString(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 ConvertColorToHexaString(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 پارمترهای پخش کننده ویدئو، از کلاس YouTubePlayerOption استفاده شده است.

نظرات خوانندگان

نویسنده: شاهین کیاست
تاریخ: ۱۳۹۱/۰۶/۲۲ ۱:۳۹

سلام ،
ممنون.

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 »
(معادل MediaElement مخصوص WPF در سیلورلایت هم هست)

نویسنده: رضا بهشتی
تاریخ: ۱۳۹۴/۰۱/۲۶ ۱۴:۳۴

سلام؛ با تشکر از مطلبتون. شما اطلاعی از چگونگی ساختن پخش زنده در وبسایت asp دارید؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۴/۰۱/۲۶ ۱۴:۳۸

« ارسال ویدیو بصورت Async توسط Web Api »

عنوان: **ساخت DropDownList های مرتبط به کمک jQuery Ajax در MVC**
نویسنده: امیرحسین مرجانی
تاریخ: ۲۱:۵۸ ۱۳۹۱/۰۷/۲۷
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net MVC, jQuery, jQuery-Tips, Ajax, MVC

در پروژه‌های زیادی با مواردی مواجه می‌شویم که دو انتخاب داریم ، و یک انتخاب زیر مجموعه انتخاب دیگر است ، مثلا 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 فعلا 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:

رابطه ای از جدول Type به جدول Category.

به کمک Scaffolding یک کنترلر برای کلاس Tap (شیر آب) می‌سازیم ، به طور عادی در فایل Create.chtml مقدار گروه را به صورت DropDown نمایش می‌دهد، حال ما نیاز داریم که خودمان DropDown را برای Type ایجاد کنیم و بعد ارتباط این‌ها را بر قرار 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 می‌کند ، ما نیز مقادیر نوع‌ها را به View مورد نظر ارسال می‌کنیم.

برای نمایش دادن هر دو DropDownList ویو مورد نظر را به این صورت ویرایش می‌کنیم :

```
<div>
            نوع
        </div>
        <div>
            @Html.DropDownList("Type", (SelectList)ViewBag.Type, "-- انتخاب ---", new { id = "rdbTyoe"
})
            @Html.ValidationMessageFor(model => model.Category)
        </div>

        <div>
            دسته بندی
        </div>
        <div>
            @Html.DropDownList("Category", (SelectList)ViewBag.Category, "-- انتخاب ---", new { id =
"rdbCategory"})
            @Html.ValidationMessageFor(model => model.Category)
        </div>
```

همان طور که مشاهده می‌کنید ، در اینجا DropDownList مربوط به Type که خودمان سمت سرور ،مقادیر آن را پر کرده بودیم نمایش می‌دهیم.

خب شاید تا اینجای کار ، ساده بود ولی میرسیم به اصل مطلب و ارتباط بین این دو DropDownList. (~~قبل از این قسمت حتما نگاهی به ساختار DropDownList یا همان تگ select بیندازید ، اطلاعات جی کوئری شما در این قسمت خیلی کمک حال شما است~~)

برای این کار ما از jQuery استفادی می‌کنیم ، **کار به این صورت است که** هنگامی که مقدار DropDownList اول تغییر کرد :
ما Id آن را به سرور ارسال می‌کنیم.
در آنجا Category هایی که دارای Type با Id مورد نظر هستند را جدا می‌کنیم
فیلدهای مورد نیاز یعنی Id و Title را می‌گیریم
و بعد به کمک Json مقادیر را بر می‌گردانیم

و مقادیر ارسالی از سرور را در optionهای DropDownList دوم (گروه‌ها) قرار می‌دهم

در ابتدا متدی در سرور می‌نویسیم ، کار این متد (همان طور که گفته شد) گرفتن Id از نوع Type استو برگرداندن Categoryهای 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 مقادیر مورد نظر را پیدا می‌کنیم و بعد به کمک Select فیلد هایی مورد استفاده را جدا می‌کنیم و به کمک Json اطلاعات را بر می‌گردانیم. توجه داشته باشید که مقادیری که در categoris قرار می‌گیرد ، شبیه به مقادیر json هستند و ما می‌تانیم 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 :

```
$('#rdbTyoe').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 مورد 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 لوپ می‌زنیم و هربار این کار را تکرار می‌کنیم.

ابتدا یک تگ option می‌سازیم

مقادیر مربوطه شامل Id که باید در attribute مورد نظر value قرار گیرد و متن آن که باید به عنوان text باشد را مقدار دهی می‌کنیم

option 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();
```

خالی می‌کنیم.

نتیجه کار برای من می‌شود این :

قبل از انتخاب :

بعد از انتخاب :

موفق و پیروز باشید

نظرات خوانندگان

نویسنده: فرشاد کریمی
تاریخ: ۱۳۹۱/۰۷/۲۷ ۲۳:۲۹

خیلی خوب و جامع بود

نویسنده: Hamid NCH
تاریخ: ۱۳۹۱/۰۷/۲۹ ۱۵:۲۶

مرسی مفید بود.

نویسنده: امیرسهیل
تاریخ: ۱۳۹۱/۰۹/۲۱ ۱۳:۲

سلام
ممنونم ، 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 متد getjson اجرا نمی‌شد.
تا قبل از متد jquery.getjson اجرا میشه. alert رو اجرا کرد. بعد دیگه اجرا نمیشه. تو firebug هم اساسا درخواستی json دیده نمیشه.
jquery-1.8.3

نویسنده: ابوالفضل
تاریخ: ۱۳۹۱/۱۰/۰۶ ۱۴:۱۰

مشکلم حل شد.
متد اجرا می‌شد. ولی در قسمت کنترلر حالت چرخشی پیش می‌اومد که نمی‌تونه به json تبدیل بشه. 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 باشه و آبجکت 2 هم متقابلا از یک مسیر دیگه به آبجکت 1 ارتباط داشته باشه، در اینجا چرخه پیش میاد و json نمی‌تونه 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 return json با اشیاء تو در تو مشکلی ندارد. فقط خروجی آن‌را 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 مونتیور شما رو مشاهده کنند. نه خطایی رو نوشتی و نه کلاس‌های مدلت مشخص بود.

نویسنده: سعید یزدانی
تاریخ: ۱۳۹۱/۱۱/۰۶ ۲۳:۷

با تشکر
میشه درباره‌ی selector این خط بیشتر توضیح بدبد

```
var option = $('<option></option>').attr("value", data[i].Id).text(data[i].Title);
```

نویسنده: امیرحسین مرجانی
تاریخ: ۱۳۹۱/۱۱/۰۷ ۱:۷

ببین دوست من ، ابتدا نگاهی به ساختار dropdown لیست‌ها داشته باش.
مقداری که باید توی dropdown نشان داده بشند چیزی به این صورته :

```
<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 ابن جا می‌خواهیم این قسمت‌ها رو خودمان درست کنیم.
در یک حلقه each به اضای مقادیری که از سرور گرفتیم loop می زنیم option را می‌سازیم و به dropdown اضافه می‌کنیم.
در مورد این قسمت هم باید بگم

```
.attr("value", data[i].Id).text(data[i].Title)
```

attr 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

مرسی از جواب کاملتون

نویسنده: ziya
تاریخ: ۱۳۹۲/۰۲/۰۶ ۱۷:۴۵

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 دسترسی پیدا کنم (یعنی چطور متوجه شم که کاربر کودوم رو انتخاب کرده که براساس انتخابش باید توی جدولهای جداگونه ذخیره بشه)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۰۶ ۲۱:۴۵

- در مورد radio button list مطلب داریم در سایت.
- برای استفاده مقدماتی از DropDownList یک مثال کامل در سایت ASP.NET هست.

نویسنده: امیرحسین مرجانی
تاریخ: ۱۳۹۲/۰۲/۰۶ ۲۱:۵۶

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 DropDown 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 این جا دو radio دارید که دارای نام یکسال هستند و مقادیر Value آن‌ها تفاوت دارد.
در اینجا هم شما می‌توانید در کنترلر خود با name ان 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: ۱۳۹۲/۰۲/۱۴ ۱۸:۴۷

سلام
من از linq, SelectListItem استفاده کردم.در view هم ازSelectListItem. ولی بقیه دقیقا مثل شما ولی جواب نمیده.مشکل چیه؟
لطفا راهنمایی کنید

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۲/۱۴ ۱۹:۳۱

یه مطلب خوب در مورد دیباگ کدهای جی‌کوئری چند وقت پیش در سایت مطرح شده. شاید بتونه به شما در پیدا کردن مشکل کمک کنه.

نویسنده: nasim

تاریخ: ۱۳۹۲/۰۲/۱۷ ۱۵:۲۹

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: ۱۳۹۲/۰۲/۱۷ ۱۶:۴۷

این قسمت jQuery.getJSON رو با Url درستی مقدار دهی کردید؟ این آدرس رو باید بر اساس برنامه خودتون تغییر بدید.

نویسنده: امیرحسین مرجانی
تاریخ: ۱۳۹۲/۰۲/۱۷ ۱۶:۵۳

سلام
اگر امکان داره دقیق‌تر سوال رو بیان کنید و یا کدی که نوشتید اینجا بگذارید.

نویسنده: nasim
تاریخ: ۱۳۹۲/۰۲/۱۸ ۱:۳۲

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>
         @{ List<SelectListItem> lstCategories = (List<SelectListItem>)ViewData["category1"]; }
            @Html.DropDownList("dbcat", lstCategories, new { id="dbcat" })
        </div>
        <div>
            @{ List<SelectListItem> lstsubCategories = (List<SelectListItem>)ViewData["Category2"]; }
            @Html.DropDownList("dbsubcat", lstsubCategories, new { id="dbsubcat"})
```

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);
        }
```

وقسمت کد جی کوئری:

```
<script src="~/Scripts/jquery-1.7.1.min.js"></script>
```

```
<script type="text/javascript">

    $('#dbcat').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); $("#dbcat").append(option);
            });
        });
   });

</script>
```

قسمت کد جی کوئری من اجرا نمیشه.حتی alert داخل

```
$('#dbcat').change(function ()
```

گذاشتم اجرا نشد.لطفا راهنمایی کنید
با تشکر

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۲/۱۸ ۹:۱۳

```
$(function () {
  // اینجا اسکریپت‌های جی‌کوئری باید تعریف شوند
});
```

قسمت کد جی‌کوئری رو باید در document ready قرار بدید

نویسنده: زرین
تاریخ: ۱۳۹۲/۰۲/۲۹ ۱۴:۲۸

سلام ،با تشکر از آموزشهای فوق العاده تون از درس1 تا این قسمت‌ها
من 2 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 متد SelectCategory رو هم فراخوانی میکنه اما dropdownList دومم رو خالی بر میگردونه!

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۲/۲۹ ۱۴:۴۸

```
Select(c => new { c.Id, c.Title })
```

خروجی شما باید همین‌ها باشه در متد انتخاب گروه، البته مطابق کدهای جی‌کوئری نوشته شده

نویسنده: زرین
تاریخ: ۱۳۹۲/۰۲/۲۹ ۱۵:۱۰

ممنونم از پاسختون؛ جواب داد (-:

نویسنده: 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

نویسنده: داود
تاریخ: ۱۳۹۳/۰۲/۲۵ ۱۱:۲۸

با سلام؛ من شبیه همچین کاری می‌خوام انجام بدم با این تفاوت که وقتی با استفاده از dropdownlistFor 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.

نویسنده: 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 رخداد 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.

نویسنده: رضا منصوری
تاریخ: ۱۳۹۳/۰۴/۱۹ ۱۴:۱۴

یک تجربه :
این سلکتور $('this).attr('value) برای کار با value در Option نال برمیگردوند و من من اینجوری استفاده کردم

```
$('#Cities').change(function () {
                jQuery.getJSON('@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 بخاطر نسخه جدیدتر Jquery هست.

نویسنده: مهیاس هانی
تاریخ: ۱۳۹۳/۱۰/۰۳ ۹:۱۱

با سلام؛ من راجب dropDown در پیج Edit.cshtml سوال دارم. چطور می‌تونم SelectedItem رو تنظیم کنم، منظورم اینکه وقتی صفحه ویرایش لود میشه، منوی کشویی روی هموم آیتمی باشه که کاربر قبلا وارد کرده و در داخل DataBase ذخیره شده.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۰۳ ۹:۲۰

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" })
```

نویسنده: مهدی محمودزاده
تاریخ: ۱۳۹۴/۰۱/۱۰ ۶:۱۶

وقت بخیر. امکانش هست دقیقا همین آموزش رو با استفاده از asp.net C# آموزش بدید؟ممنون میشم.

عنوان: **استفاده از دکمه‌های CSS توئیتر در ASP.NET MVC - قسمت دوم**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۲۸ ۲۱:۲۳
آدرس: www.dotnettips.info
گروه‌ها: MVC

در قسمت اول مشاهده کردیم که به چه صورت می‌توان از دکمه‌های CSS ایی 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 اعتبارسنجی موفقیت آمیز بوده یا خیر؛ سپس دکمه را غیرفعال کنیم.
برای پیاده سازی این امر باید jQuery Validator همراه با ASP.NET MVC 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 متد جاوا اسکریپتی validateThisForm، سیستم jQuery Validator به صورت دستی فراخوانی شده و سپس نتیجه عملیات بازگشت داده می‌شود.
در متد 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')" class="btn btn-
info">ارسال</span>
    </fieldset>
```

```
}
```

دریافت کدهای کامل این مثال:

MvcApplication27.7z

عنوان: با ASP.MVC چه مزایایی را به دست خواهیم آورد
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۸/۰۴ ۱۲:۸
آدرس: www.dotnettips.info
گروه‌ها: MVC

عموما 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 مسایل 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 پیش فرض از سیستم Routing موجود در زیر ساخت ASP.NET برای نمایش Urlهایی 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 ... عموما 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 الزاما 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 متدی ارجاع داده شود.

این مساله از لحاظ امنیتی نیز مهم است. درخواست‌های رسیده به MVC سبب اجرای هیچ فرمی نخواهند شد. درخواست‌ها حتما باید از فیلتر یک کنترلر عبور کنند تا اجرایی شوند.

3- امکان مدیریت بهتر قسمت‌های مختلف سایت در پوشه‌های جداگانه

اگر به سورس اکثر سایت‌های مبتنی بر ASP.NET Web forms توجه کنیم، تمام فایل‌های آن‌ها در ریشه سایت قرار دارند منهای فایل‌های CSS و JS و تصاویر. در ASP.NET MVC از ابتدای کار، هر قسمت از سایت در پوشه‌های 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 توسط 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 متداول 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 دهید. به یک فرم یا View کلاسی را انتساب خواهید داد. فریم ورک خواص آن‌را به صورت خودکار در حین ارسال به سرور مقدار دهی خواهد کرد. این مورد حتی در حین کار با jQuery Ajax نیز صادق است.

این مساله کار با ORMها را نیز ساده‌تر می‌کند. از این جهت که تمام آن‌ها نهایتا 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 MVC

از آنجائیکه فرم‌های ASP.NET Web forms از نوع strongly typed نیستند (در دات نت 4 و نیم اندکی بهبود در حد گریدهای 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 دهند، 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 مجددا 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 اعتبارسنجی آن‌را غیرفعال کنیم، مثلا برای دریافت html از کاربر، نیاز است کلا آن‌را 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 ایی را دریافت کند، غیرفعال کرد؛ نه برای کل فرم و نه در سطح کل برنامه.

9- امکان استفاده از فرم‌های و Viewهای 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 Razor است که تمام برنامه نویس‌های MVC پس از مدتی به روان بودن و طراحی طبیعی و عالی آن اعتراف 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 کنترلرها اعمال کرد و به صورت خودکار توسط فریم ورک پیش یا پس از اجرای یک متد اجرا می‌شوند. به این ترتیب حجم قابل ملاحظه‌ای از if و elseها را می‌توان در این فیلترها کپسوله کرد و کدهای متدهای کنترلرها را تمیزتر و زیباتر نمود.

12- سازگاری کامل با jQuery و jQuery 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 spanها جایگزین کرده و توسط فریم ورک‌های CSS ایی 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 Ajax نیازی نیست تا متدی را static تعریف کنید و به این ترتیب از مزایای امنیتی توکار ASP.NET محروم شوید (مثلا 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 Ajax نسبت به فریم ورک Ajax همراه با ASP.NET web forms بسیار سبک‌تر و سریعتر عمل می‌کند چون نیازی ندارد تا هر بار 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 MVC 4 سیستم bundling آن از نمونه مشابه موجود در وب فرم‌ها کامل‌تر است و جهت فشرده سازی و یکی کردن هر دو مورد فایل‌های css و js می‌تواند بکارگرفته 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 و ORMها نیز نهایتا 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 فریم ورک وب فرم‌ها و MVC امکان تزریق وابستگی‌ها وجود دارد. اما در MVC می‌توان در میانه کار وهله سازی کنترلرها، دخالت کرد و کنترل آن را کاملا 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 جزئی از فریم ورک 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 providerهای سفارشی.

و یا برای نمونه Razor به صورت پیش فرض امن طراحی شده است. خروجی Razor همواره و در بدو امر، html encoded است مگر اینکه برنامه نویس آگاهانه آن‌را 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 referrer فرم ارسال به سرور را درنظر بگیرید. در وب فرم‌ها می‌توان این‌کار 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 2 سالی هست با سی شارپ و Win App کار کردم و میخوام 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 مستقیما از ASP.NET MVC شروع کنم. البته به صورت مقدماتی با ASP.NET آشنا هستم. ممنون میشم راهنمایی کنید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۸/۰۴ ۱۶:۳۸

مستقیما می‌تونید با MVC شروع کنید. البته پیشنیازهای HTML، CSS و جاوا اسکریپت را هم باید لحاظ کنید.

نویسنده: امیرحسین مرجانی
تاریخ: ۱۳۹۱/۰۸/۰۵ ۲۳:۴۹

واقعا 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 قبلا هم 16 مورد از کاربردها و مزایای sliverlight 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 کننده ای دریافت نکرده ام.

چرا با وجود کاربردها و مزایای silverlight (مخصوصا موارد 6-11-13-16 در لینک فوق) هنوز mvc asp.net مطرح تر، بازار کار بیشتر و حتی طرفداران بیشتری در بین برنامه نویسان دارد؟ (صرف نظر از افزونه ای که برای مرورگرها دارد)

لطفا 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 (اینترانت) و نه وب سایت‌های عمومی. خصوصا از لحاظ بحث SEO که سیلورلایت اساسا 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 برنامه‌های به شدت امن قابل قبول نیست. اما در یک شبکه داخلی شاید این مساله اهمیتی نداشته باشد.
- تیم مونو اخیرا 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 تکذیبه صادر شد اما الان تمام کسانی رو که قبلا فقط در مورد سیلورلایت مطلب می‌نوشتند (اعضای سابق این تیم)، صرفا در مورد فریم ورک‌های جدید js 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.

نویسنده: M.Q
تاریخ: ۱۳۹۱/۰۸/۰۹ ۱۴:۲۹

ممنون از شما جناب نصیری

به امید روزی که برنامه نویسی وب تنها با دانش یک تکنولوژی برای برنامه نویسان میسر شود.

نویسنده: sysman
تاریخ: ۱۳۹۱/۱۲/۰۳ ۱۹:۱۰

نوشتن یک برنامه enterprise با استفاده از سیلورلایت سریعتر و راحت‌تر است یا mvc؟ با توجه به عدم ارایه سیلورلایت ۶ آیا نوشتن چنین برنامه هایی که حداقل دارای طول عمری 10 ساله هستند کار عاقلانه ای است؟ یا به طور کلی بهتره بپرسم آیا در حال حاضر گزینه دیگری به جز سیلورلایت برای نوشتن یک برنامه Lob وجود دارد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۰۳ ۲۰:۳۳

- باتوجه به جمله بندی که بکار بردید، به نظر تصمیم خودتون رو در مورد سیلورلایت گرفتید. اما اگر برای 10 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 10 سال بعد نخواهد بود.
سیلورلایت و فلش با تبلیغ منفی استیوجابز به علت مشکلات فلش در پلتفرم آن‌ها از بین رفتند.
- سریعتر و راحت‌تر برای برنامه‌ای که قرار است 10 سال کار کند معنا ندارد. چون یک چنین سیستمی حداقل 2 سال طول خواهد کشید تا نگارش یک آن جا بیفتد و بعد کار نگهداری خواهید 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 رو 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 می‌تونید 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. این مورد جایگزین بهتری است برای سیلورلایت.

نویسنده: sysman
تاریخ: ۱۳۹۱/۱۲/۰۵ ۱۲:۱۱

ممنون از راهنمایی شما. هنوز تصمیم قطعی گرفته نشده. در حقیقت چند تستی که در خصوص سیلورلایت انجام دادم کمی من رو ناامید 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 بتونه 10000 کاربر همزمان رو جوابگو باشه. آیا شما تجربه عملی با این حجم کاربر دارید؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۰۵ ۱۲:۱۸

برای مدیریت 10 هزار کاربر، یک سری روش‌هایی در ویندوزهای سرور وجود دارند مانند clustering ، load balancing و امثال 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: ۱۳۹۳/۰۸/۲۳ ۰:۴۸

ممنون از شما بابت اطلاع رسانی خوبتون مثل همیشه واقعا asp.net با mvc تغییرشگرفی کرده
من همون اول که متوجه شدم از یکی بودن form اون هم runat=server رهایی پیدا کردیم خیلی خوشم اومد
این مسئله همیشه موجب آزار بود و هر دفعه باید یه جوری باهاش سر میکردیم و البته viewstate رو هم نباید فراموش کرد

عنوان: **نحوه اضافه کردن Auto-Complete به جستجوی لوسین در ASP.NET MVC و Web forms**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۸/۰۶ ۹:۵۸
آدرس: www.dotnettips.info
گروه‌ها: Lucene.NET, ASP.Net, MVC

پیشنیازها:

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 شده‌است. در مطلب جاری به بررسی این مورد به همراه دو مثال Web forms و MVC پرداخته خواهد شد.

قسمت عمده مطلب جاری با پیشنیازهای یاد شده فوق یکی است. در اینجا فقط به ذکر تفاوت‌ها بسنده 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 برجسته 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 Boost به لوسین 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 آن‌را 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 SearchResultها 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 نمائید lu 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, autoFill: 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.
return Content هم سبب بازگشت این اطلاعات به افزونه خواهد شد.
- کدهای View متناظر بسیار ساده هستند. تنها نام TextBox تعریف شده مهم می‌باشد که در متد جاوا اسکریپتی EnableSearchAutocomplete استفاده شده است. به علاوه، نحوه مقدار دهی آدرس دسترسی به اکشن متد ScoredTerms نیز مهم می‌باشد.
- در متد EnableSearchAutocomplete نحوه فراخوانی افزونه autocomplete را ملاحظه می‌کنید.
جهت آن، به راست به چپ تنظیم شده است. با 2 کاراکتر ورودی فعال خواهد شد با وقفه‌ای کوتاه. نیازی نیست تا انتخاب کاربر از لیست ظاهر شده حتما با عبارت جستجو شده صد در صد یکی باشد. حداکثر 20 آیتم در لیست ظاهر خواهند شد. اسکرول بار لیست را حذف کرده‌ایم. عرض آن به 300 تنظیم شده است و نحوه فرمت دهی نمایشی آن‌را نیز ملاحظه می‌کنید. برای این منظور از متد 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 متد result مقدار دهی شده).
window.location را به عنصر دوم آرایه row هدایت خواهیم کرد. این عنصر دوم مطابق کدهای اکشن متد تهیه 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. خصوصا توضیحات مرتبط با متد 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
    {
        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 خواهند 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 متد MapPostToDocument ، فیلد دیگری را به نام مثلا 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 کردم 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 برگردونم 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؟
مثلا نتایج زیر همون تکست باکس نمایش داده بشه.
آیا باید نوع اکشن چیز دیگه ای باشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۰/۲۵ ۱۴:۲۱

نتیجه جستجوی لوسین مثلا در مطلب فوق لیستی از SearchResultها است. همین لیست را در اکشن متد بازگشت دهید. سپس در View متناظر 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 اطلاعتتون رو در اختیار دیگران قرار میدید بنده از کدهای بالا تویه سایتم استفاده کردم و جستجوی سایت مثل ساعت داره کار میکنه فقط تویه ظاهرش به مشکل برخوردم بنده میخوام اون box رو که اطلاعات توش نمایش داده میشه از سمت راست تکس باکس تراز بشه مثل همون عکسی که بالا گذاشتید از بنده 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 بنده position:absolute هستش نمیدونم ماله اینه که نمایش نمیده یا مثلا 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.
مثلا 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 مشکلی 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. می‌تونید اینجا متن انتخابی رو به صورت مثلا یک کوئری استرینگ تعریف کنید و بعد در صفحه‌ای دیگر دریافت و نمایش بدید.

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 داره 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 خواهید 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 کند و کدام قسمت‌ها باید حذف و اضافه شوند
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۲۱ ۲۰:۳۰

لطفا اولین پیشنیاز عنوان شده را مطالعه کنید.

نویسنده: میهمان
تاریخ: ۱۳۹۱/۱۱/۲۲ ۰:۱۷

با سلام مجدد؛ تمامی پیشنیازهای 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 پیشنیاز عنوان شده فرموده بودید " کتابخانه Lucene مستقل است از منبع داده مورد استفاده و تنها اطلاعاتی با فرمت شیء Document معرفی شده به آن‌را 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 رکرودها 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 ۰:۵۱

- 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 می‌ده.
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 ویکی پدیا)، اما جستجوی آن فوق العاده سریع و با کیفیت بود. این ویدیو رو در اینجا می‌تونید مشاهده کنید:
Full-text search with Lucene and neat things you can do with it

نویسنده: morteza
تاریخ: ۱۳۹۱/۱۲/۰۱ ۲۲:۳۰

سلام آقای نصیری میخواستم ببینم آیا شما اطلاعی از نحوه غیر فعال کردم اینتر تویه این پلاگین دارید بنده میخوام وقتی کاربر اینتر رو فشار داد جستجوی معمولی انجام بشه و اگه خواست از مواردی که توسط autocomplate براش آمده توسط موس یکی رو انتخاب کنه و به صفحه مربوطه هدایت بشه ممنون.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۰۱ ۲۲:۴۹

کلید enter با case KEY.RETURN در این افزونه جی‌کوئری مدیریت می‌شود.

نویسنده: علیرضا پایدار
تاریخ: ۱۳۹۱/۱۲/۲۰ ۱۰:۵۵

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

ممنون

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 نباشید
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. مرسی

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 پیشنیازها که کار با لوسین رو گفته بودین توابعی مثله SearchByPartialWords و Query و غیره ایجاد شده ولی تو اینجا یه جور دیگه جستجو کردین توی ایندکس‌ها . میخواستم بدونم او توابع رو مگه واسه سرچ تو ایندکس‌ها ننوشتین؟ تازه اینجا شما دارین متن Title رو میفرستین سمت کاربر . این highlight کردن عبارت جستجو شدش پس چی شد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۷/۲۷ ۱۱:۳۵

- این هم یک روش دیگر هست. از آن روش‌های ذکر شده در پیشنیازها هم می‌توانید استفاده کنید.
- highlight کردن عبارت جستجو شد ه 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 نوشتین یه دونه FastVectorHighlighter ایجاد کردین . اگه highlight توسط auto-complete خودکار انجام میشه پس نقش این چیه؟ میشه بگین کدوم روش بهتره ؟ یا فرقی نمیکنه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۷/۲۷ ۱۱:۵۲

دو بحث وجود داره:
- برجسته سازی قسمتی از عبارت جستجو شده در لیست نمایش داده شده توسط افزونه auto-complete. این مورد خودکار است و توسط افزونه انجام می‌شود.
- برجسته 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؟
مثلا یه Dropdown داشته باشیم وبخواییم مثلا مقدار categoryid رو هم پاس بدیم تا به query مربوطه اضافه بشه.من dropdown رو اضافه کردم اما مقدارش رو نمیشه در هندلر مربوط به Search دریافت کرد مثل q.

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 مثلاً یه عکس بهش اضافه کنم یا یه جدول ساخته شده با HTML رو توش 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 متد 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 مشکلی نیست ولی اگر صفحه در یک پوشه دیگر قرار گیرد جواب نمی‌ده. توضیح اینکه از متد ResolveClientUrl هم استفاده کردم ولی جواب نداد :

```
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: محمّد 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 کدوم جدوله. حالا راهی هست که به جز Id فیلد دیگه ای هم برای بروزرسانی یا حذف یک سند ارسال کرد؟ مثلا 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

متد DeleteDocuments 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);
```

نویسنده: آرش مصیر
تاریخ: ۱۳۹۳/۱۲/۲۸ ۱۲:۱۹

با سلام
آیا در حال حاضر این روش همچنان روش بهینه و مناسبی برای جستجوی auto-complete است؟یا روش‌های جدیدتری هم وجود دارند؟(تاریخ انتشار این کتابخانه 2012 بوده است)
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۲/۲۸ ۱۲:۲۴

بله. نسخه‌ی 4 آن هنوز نهایی نشده‌است: « نگاهی به وضعیت کتابخانه‌ی Lucene.NET »

عنوان: **مقابله با پسوردهایی که ساده حدس زده می‌شوند**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۸/۱۰ ۲۱:۵
آدرس: www.dotnettips.info
برچسب‌ها: MVC, Security, Validation

هر ساله لیستی از پرکاربردترین کلمات عبور کاربران در دنیا، منتشر می‌شود که یک نمونه از آن‌را در اینجا می‌توان مشاهده کرد:
« Splashdata، توسعه دهنده نرم افزارهای امنیتی، فهرست سالانه خود را از رایج‌ترین رمزهای عبور منتشر کرده است. »

می‌شود 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 می‌توان 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 euqalCharsLen = data.ToCharArray().Count(x => x == firstElement);
            if (euqalCharsLen == data.Length) return true;
            return false;
        }
    }
}
```

متد الحاقی IsSafePasword فوق بررسی می‌کند که آیا کلمه عبور انتخابی:

- خالی نیست.
- بیشتر از 5 کاراکتر 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 = "کلمه عبور")]
        [Required(ErrorMessage = "لطفا کلمه عبور خود را وارد نمائید")]
        [DataType(DataType.Password)]
        [StringLength(50, MinimumLength = 5, ErrorMessage = "حداقل طول کلمه عبور 5 حرف است")]
        [Remote(action: "CheckPassword",  controller: "Register",  HttpMethod = "POST",
                    ErrorMessage = "کلمه عبور وارد شده را راحت می‌توان حدس زد!")]
        public string Password1 { get; set; }
    }
}
```

عنوان: **غیرفعال کردن کش مرورگر در MVC**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۸/۱۱ ۲۱:۵
آدرس: www.dotnettips.info
برچسب‌ها: MVC

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 متد 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 مواقع 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)
```

عنوان: Best Practice هایی برای ASP.NET MVC - قسمت اول
نویسنده: شاهین کیاست
تاریخ: ۱۳۹۱/۰۸/۱۲ ۱۸:۵
آدرس: www.dotnettips.info
برچسب‌ها: ASP.Net, MVC, Tips

در سایت جاری مطالب زیادی درباره ASP.NET MVC نوشته شده است. این مطلب و قسمت بعدی آن مروری خواهد داشت بر Best Practiceها در ASP.NET MVC.

استفاده از NuGet Package Manager برای مدیریت وابستگی‌ها

درباره اهمیت NuGet برای مصرف کنندگان قبلا 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؛ مثلا تیم‌های برنامه نویسی می‌توانند کتاب خانه‌های مشترک خودشان را در مخزن‌های سفارشی NuGet قرار دهند و توزیع و Versioning آن‌را به NuGet بسپارند.

تکیه بر Abstraction (انتزاع)

Abstraction در طراحی سیستم‌ها منجر به تولید نرم افزار هایی Loosely coupled با قابلیت نگهداری بالا و همچنین فراهم شدن زمینه برای نوشتن Unit Test می‌شود.
اگر به مطالب قبلی وب سایت برگردیم در مطلب چرا ASP.NET MVC گفته شد که :
2) دستیابی به کنترل بیشتر بر روی اجزای فریم ورک :
در طراحی ASP.NET MVC همه‌جا 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 routing engine، view 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 آن‌را دوست ندارید؟ آن‌را 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 نوشتن آزمون‌های واحد.

از کلمه‌ی کلیدی New استفاده نکنید

هر جا ممکن است کار وهله سازی اشیاء را به لایه و حتی Framework دیگری بسپارید. هر زمان اشیاء نرم افزار خودتان را با کلمه‌ی new وهله سازی می‌کنید اصل Abstraction را فراموش کرده اید. هر زمان اشیاء نرم افزار را مستقیم وهله 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 مستقیما استفاده نکنید (از HttpContextBase استفاده کنید)

از .NET 4 به بعد فضای نامی تعریف شده که در بر دارنده‌ی کلاس‌های انتزاعی (Abstraction) خیلی از قسمت‌های اصلی ASP.NET است. یکی از مواردی که در توسعه‌ی ASP.NET معمولا زیاد استفاده می‌شود، شیء HttpContext است . استفاده از HttpContextBase را به استفاده از HttpContext 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 (مثلا : دشوار است که از روی نام یک ID مشخص کنم این ID چطور به ID دیگری مرتبط است و یا اصلا ربط دارد یا خیر)
با اشتباهات املایی یا عدم رعایت حروف بزرگ و کوچک ایجاد مشکل می‌کنند.
به Refactoring 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>
```

مثلا مثال دوم مشکلات رشته‌های جادویی را ندارد.
در رابطه با Magic strings این مطلب 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 کنترلر ها) البته در نظر داشته باشید که قطعا تولید HTML در متدهای کمکی کلاس هایی 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 کاملا 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 را 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. مثلا:

```
@if(Model.IsAnonymousUser) {
    <img src="@Url.Content("~/content/images/anonymous.jpg")" />
} else if(Model.IsAdministrator) {
    <img src="@Url.Content("~/content/images/administrator.jpg")" />
```

```
} else if(Model.Membership == Membership.Standard) {
    <img src="@Url.Content("~/content/images/member.jpg")" />
} else if(Model.Membership == Membership.Preferred) {
    <img src="@Url.Content("~/content/images/preferred_member.jpg")" />
}
```

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 (بخصوص در مورد استفاده از best practiceها) وضعیتی است که اصطلاحاً بهش overkilling میگن (اگر اشتباه نکنم) ، یعنی بعضی وقتا 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 عفیفی
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 می‌کنم.
چرا همینطوره.
کدها همه مثال هستند و هدف این مطلب چیز دیگری بوده.

عنوان: **چک لیست تهیه یک برنامه ASP.NET MVC**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۸/۱۹ ۱۱:۵
آدرس: www.dotnettips.info
گروه‌ها: MVC

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 ViewData ، ViewBag و امثال آن استفاده نشده و به ازای هر View یک مدل متناظر (ViewModel) ایجاد شود.
- فایل پروژه برنامه توسط یک ادیتور متنی ویرایش شده و MvcBuildViews آن به True تنظیم شود.
- مدل‌های متناظر با جداول بانک اطلاعاتی نباید مستقیما در Viewهای برنامه استفاده شوند.
- پوشه 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.
- از سیستم minification و bundling ، برای یکی سازی اسکریپت‌ها و CSSهای برنامه استفاده شود.
- قسمت custom errors فایل web.config برنامه به نحو صحیحی مقدار دهی شود.
- تمام فرم‌های عمومی برنامه باید دارای AntiForgeryToken باشند.
- تمام فرم‌های عمومی برنامه باید captcha داشته باشند.
- پوشه‌های Content و Scripts از سیستم مسیریابی تعریف شده در Global.asax خارج شوند.
- MvcHandler.DisableMvcResponseHeader = true به Application_Start اضافه شود.
- اگر فقط از Razor به عنوان ViewEngine استفاده می‌شود، در Application_Start، باید سایر ViewEngineهای مورد استفاده، حذف شوند .
- فیلتر پیش فرض مدیریت خطاها حذف و بجای آن از ELMAH استفاده شود.
- در web.config، مقادیر executionTimeout و maxRequestLength مرتبط با httpRuntime تنظیم شوند. همچنین enableVersionHeader آن نیز خاموش گردد.
- 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)
- در هیچ کنترلری نباید جزئیات پیاده سازی متدی مشاهده شود. تمام پیاده سازی‌ها باید به لایه سرویس‌های مختلف برنامه منتقل و از طریق تزریق وابستگی‌ها در دسترس باشند.
- اگر نیاز به مشخص سازی آدرسی در سایت است (خصوصا در اسکریپت‌ها) باید از Url.Action استفاده شود و نه رشته‌ها.
- بهتر است بومی سازی برنامه از روز اول آن درنظر 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 کتابخانه WebActivator استفاده شود.
- امکان دوبار کلیک کردن بر روی تمام دکمه‌ها نباید وجود داشته باشد.
- از هش‌های ترکیبی استفاده شود. مستقیما 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، نباید بتواند از سایت استفاده کند.
- کلیه تغییرات تنظیمات و محتوای مهم سایت باید برای مدیر سایت بلافاصله ایمیل شوند.
- یک سری کارهای متداول مانند تهیه فایل‌های robots.txt، crossdomain.xml، apple-touch-icon-XxY.png، favicon.ico و sitemap.xml (ترجیحا 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 ایی استفاده شود .
- در تمام قسمت‌هایی که فایلی آپلود می‌شود باید بررسی شود فایل‌های نا امن (فایل‌های اجرایی ASP.NET) قابل آپلود نباشند .
- حین کار با بانک‌های اطلاعاتی یا از ORM استفاده شود و یا از کوئری‌های 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 لاگین سایت، autocomplete خاموش شود.
- تمام deleteهای برنامه باید به HttpPost محدود شوند. تمام گزارشات و نمایش اطلاعات غیرعمومی برنامه به HttpGet محدود شوند.
- تعداد رفت و برگشت‌های به بانک اطلاعاتی در یک صفحه توسط پروفایلرها بررسی شده و اطلاعات عمومی پرمصرف کش شوند .
- در هیچکدام از کلاس‌های ASP.NET MVC نباید از HttpContext مستقیما 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 واقعا مرتبط هستند به سایت جاری و آیا واقعا Ajax ایی 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 Helperهای 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.

- MvcHandler.DisableMvcResponseHeader = true به Application_Start 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 (کار هایی 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 رو باید بیشتر کنید. همچنین حجم فایل که بالا رفت نیاز است سرور زود timeout نزند.
- خاموش کردن enableVersionHeader و همچنین MvcHandler.DisableMvcResponseHeader سبب می‌شود تا رباط‌های باگ یاب سایت‌ها نتوانند دقیق عمل کنند چون شماره نگارش MVC و ASP.NET مورد استفاده مخفی می‌شود.
- از کوکی استفاده کنید. برای شروع: (^)
در دوران ASP کلاسیک، از سشن عموما برای مشخص سازی وضعیت لاگین 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 مقاله ای 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؟ باتشکر فراوان.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۸/۲۰ ۲۳:۱۲

WebGrid توکار آن خوب است: (^)
پروژه MVC Contrib هم گرید دارد: (^)
به علاوه دست شما در MVC باز است. مثلا از jqGrid هم می‌تونید استفاده کنید: (^)

نویسنده: ایلیا اکبری فرد

تاریخ: ۱۳۹۱/۰۸/۲۰ ۲۳:۱۵

سپاس فراوان.
شب خوش.

نویسنده: امیرحسین مرجانی
تاریخ: ۱۳۹۱/۰۸/۲۲ ۱۹:۰

سلام آقای نصیری

با این مطلب دیگه حجت رو بر ما تموم کردی.

واقعا ممنونم خیلی عالی بود

اگر منابع انگلیسی از نوع همین چک لیست می‌شناسید لطفا معرفی کنید؟

نویسنده: کیارش سلیمان زاده
تاریخ: ۱۳۹۱/۰۸/۲۲ ۲۳:۲۳

سلام
" 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) "

در این صورت:
- Captchaها که از سشن استفاده میکنن دیگه کار نمیکنن!راه حل منطقی چیه؟
- TempData کار میکنه یا این امکان هم از کار میفته؟
ممنون.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۸/۲۲ ۲۳:۵۸

- از Captchaایی 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 StorageProvider **Cookie** دارد و هم SessionStorageProvider.
- نیازی به TempData نیست و با استفاده صحیح از امکانات Routing می‌شود آن‌را 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 کلا در مورد استفاده از امکانات پیشرفته سیستم Routing بی‌خبرند.

نویسنده: صابر فتح اللهی
تاریخ: ۱۳۹۱/۰۸/۲۳ ۱:۸

سلام
می تونی سبد خرید توی Profile نگهداری کنی، با قطع برق و اتصال اینترنت و ... راحت قابل بازیابی هست.

نویسنده: محمد صاحب
تاریخ: ۱۳۹۱/۰۸/۲۳ ۱۴:۳۳

راه معادل پیاده سازی سبد خرید.

نویسنده: sdg
تاریخ: ۱۳۹۱/۰۸/۲۴ ۱۵:۲۵

مورد سوم لینک MvcBuildViews درسته آیا ؟!

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۸/۲۴ ۱۶:۱۲

بله. مطلب انتهای لینک داده شده را تحت عنوان «نکته‌ای تکمیلی در مورد Strongly typed viewها»، مطالعه کنید:
«... پاسخ: بله. باید فایل پروژه را اندکی ویرایش کرده و مقدار MvcBuildViews را که به صورت پیش فرض false هست، true نمود...»

نویسنده: امیرحسین مرجانی
تاریخ: ۱۳۹۱/۰۸/۳۰ ۲:۲

سلام
برای مدیریت سبد خرید حتما به پروژه MusicStore رو ببینید.

خیلی می‌تونه بهتون کمک کنه.

نویسنده: حسین
تاریخ: ۱۳۹۱/۰۹/۱۱ ۱۱:۳۵

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 فرم هایی که کاربر اطلاعات وارد میکنه مثل Create و Edit و ...
ممنون.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۹/۱۱ ۱۱:۴۵

- در فرم‌هایی که اطلاعاتی را به سرور Post می‌کنند الزامی است (خصوصا از لحاظ مسایل امنیتی)
- در گزارشات ... «بهتر» است اعمال شود. مثلا 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 صرفا مصرف کننده حافظه خواهند شد اگر از viewModel استفاده نکنید. (ولی ... در کل بهتر است اینگونه باشد)

نویسنده: کیارش سلیمان زاده
تاریخ: ۱۳۹۱/۱۱/۰۹ ۱۶:۵

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؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۰۹ ۱۶:۳۸

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** می نویسیم ،مثلاً:

```
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;
}
```

-1اگه این روش درسته،حالا:
- برای هر متد که نیازه یه سری فیلد مورد نیاز رو برگردونه، **یه کلاس جداگانه** باید تعریف کرد؟(در اینجا UserInfoViewModel)؟

- این کلاس UserInfoViewModel باید جز ViewModelهای لایه UI باشه و در لایه Models قرار بگیره؟
در اینصورت (کلاس UserInfoViewModel باید جز ViewModelهای 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؟
-2اگه 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 داره. 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 تغذیه 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 UserInfoViewModel در قسمت ViewModel درسته؟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 می‌تونه domain model بازگشت بده و هم view model. در عموم کارهای مایکروسافت این دو بجای هم بکار رفتن و یکی هستند. ما اینجا جداشون 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 domain model یا DTO برگرداند و در کنترلر با استفاده از automapper یا مپینگ دستی viewmodel ساخته شود و به view ارسال شود.

هرچند طراحی کاملا سلیقه ای 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. ViewModelها در MVC یک سری کلاس ساده دارای خاصیت ... خاصیت ... خاصیت بیشتر نیستید.
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 View وجود داشته باشد یا خیر. کاملا 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، خصوصا در گزارشات. این مساله سبب می‌شود که شما 20 فیلد را بازگشت داده و سپس در سمت کلاینت 3 مورد آن‌را 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 آن‌را به ViewModel نگاشت کنید. این نگاشت بهتر است همان ابتدای کار در select تهیه شده صورت گیرد؛ به این ترتیب واکشی از دیتابیس مثلا 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
عموما برای سادگی کار و تفکیک پذیری یک برنامه MVC پروژه به لایه هایی تقسیم بندی می‌شود لطفا در مورد لایه‌های زیر یک توضیح کلی بدهید و اینکه معمولا در این لایه‌ها چه قسمت هایی از پروژه قرار می‌گیرند :

Application.Web
Application.Manager

Application.Repository

Application.DAL

Application.DTO

Application.Core

Application.Common

و اینکه لایه‌های DomainClasses و DataLayer در دل این لایه‌ها قرار می‌گیرند یا بصورت لایه‌های 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؟
متشکرم.

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 میبنده به خطا! فرضاً میخوام AccountModels.cs پیش فرض رو منتقل کنم به class library جدید؛ به مشکل رفرنس‌ها برمیخورم: ویژوال استودیو 2012 و پروژه MVC 4

ضمناً یه سوال دیگه هم از خدمت شما داشتم، بنظرتون درسته بریم سمت MVC 4 یا فعلاً برامون در انتشار پروژه مشکل ایجاد میکنه؟ مثلاً هاست‌های کمی ممکنه دات نت 4.5 یا MVC 4 رو ساپورت کنن و ...
ممنون از زحماتی که میکشید، مخصوصاً از این چک لیست کامل ...

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۱۹ ۱۲:۳۱

- حالت پیش فرض Target Framework Versionپروژه‌های دات نت 4 بر مبنای Client profile است. یعنی دسترسی به امکانات وب در آن وجود ندارد. برای اصلاح آن به خواص پروژه مراجعه کرده و Client profile یاد شده را به Full profile تغییر بدید تا بتونید ارجاعات لازم به اسمبلی‌های مرتبط با پروژه‌های 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 4 تهیه شده و نه دات نت 4 و نیم. به این معنا که به راحتی حتی بر روی ویندوز سرور 2003 سرویس پک 2 (و ویندوزهای بعدی) قابل نصب است.

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۱/۱۱/۲۵ ۱۶:۵۴

با سلام.
امکات دارد که برای قسمت " امکان دوبار کلیک کردن بر روی تمام دکمه‌ها نباید وجود داشته باشد " مثالی بزنید؟
باتشکر.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۲۵ ۱۸:۳

« استفاده از دکمه‌های CSS توئیتر در ASP.NET MVC - قسمت دوم »

نویسنده: حسام محمدی
تاریخ: ۱۳۹۱/۱۱/۲۷ ۱۸:۷

سلام. میخواستم بدونم کار درستیه که کلاس‌ها و اینترفیس‌های مربوط به Service Layer (برای کار با دیتابیس) رو داخل DomainClasses بذاریم یا بهتره که یک پروژه دیگه برای لایه سرویس درست کرد؟

نویسنده: 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 کلاس‌های سرویس ، DomainClasses و ... را در همان پروژه اصلی داشته باشیم(منظور همان پروژه‌ای 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 ماژول 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 iis 7.5 استفاده می‌کنم ولی gzip فعال نیست و وقتی با وب سرور تماس می‌گیرم میگه فقط برای سرورهای لینوکس به ظور پیش فرض فعال هست!.
ممنون میشم کمکم کنید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۰۹ ۹:۳۷

من یک مقدار کدهاش رو برای اعمال تنظیمات تغییر دادم: HttpCompress.zip

نویسنده: حسین
تاریخ: ۱۳۹۲/۰۲/۱۵ ۱۷:۳۵

سلام. در مورد " تمام گزارشات و نمایش اطلاعات غیرعمومی برنامه به HttpGet محدود شوند " یعنی برای actionها حتماً [HttpGet] رو ذکر کنیم؟ چون به صورت پیش فرض اگر ذکر نکنیم هم HttpGet درنظر گرفته خواهد شد.
ممنون.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۱۵ ۱۷:۵۳

نه. اگر ذکر نشه به درخواست‌های از نوع Head هم مثلا 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 قبلا 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 نباشید
آقای نصیری بی زحمت در مورد:
در هیچ کنترلری نباید جزئیات پیاده سازی متدی مشاهده شود. تمام پیاده سازیها باید به لایه سرویسهای مختلف برنامه منتقل و از طریق تزریق وابستگیها در دسترس باشند.
بیشتر توضیح دهید.
متشکرم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۱۸ ۱۳:۳۳

مراجعه کنید به دوره « بررسی مفاهیم معکوس سازی وابستگی‌ها و ابزارهای مرتبط با آن »

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۲/۰۴/۲۷ ۲۰:۵

با سلام.
آیا می‌شود قابلیت این کتابخانه را در کل پروژه بصورت خودکار فعال کرد؟ سپاس.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۲۷ ۲۰:۲۱

HTTP Module 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 ماژول استفاده می‌کند.

نویسنده: اکبر
تاریخ: ۱۳۹۲/۰۷/۰۷ ۷:۳۹

با سلام.
من طبق راهنمایی شما ماژول سشن را از وب کانفیگ حذف کردم، حالا رویداد سشن استارت من کار نمیکند، چون محاسبه آمار افراد آنلاین را در این رویداد قراردارد، در مورد محاسبه تعداد افراد آنلاین کمی راهنمایی میکنید؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۷/۰۷ ۱۰:۴۵

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، کسانی‌را که در 5 دقیقه قبل رکورد به روز شده‌ای نداشته‌اند از این دیکشنری حذف کنید.

نویسنده: ایمان اسلامی
تاریخ: ۱۳۹۲/۰۷/۲۹ ۹:۴۵

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 interval 5 دقیقه ای.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۷/۲۹ ۱۰:۳۶

- Request.IsAuthenticated یک IP را در روال رخداد گردان AuthenticateRequest و یا PostAuthenticateRequest بررسی 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

تاریخ: ۱۳۹۲/۰۷/۲۹ ۱۲:۱۵

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 بخاطر سپاری کلمه عبور) کاربر را بلافاصله 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 واقعا تفاوتی وجود دارد یا نه. اصلا مهم است یا نه.
من در این سایت این تعداد عدد رو برای 15 دقیقه کش می‌کنم. یعنی اصلا به روز نیست و لحظه‌ای محاسبه نمیشه و نمایش داده نخواهد شد. چرا؟ چون یک کاربرد مهم و اساسی نیست در سایت. فقط یک نمایش کلی است. فقط یک حس 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.قبلا 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.ضمنا ممنون می‌شم موارد زیر را با مثال توضیح دهید
«-مدل‌های متناظر با جداول بانک اطلاعاتی نباید مستقیما در Viewهای برنامه استفاده شوند.
- پوشه 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 متدی مشاهده شود. تمام پیاده سازی‌ها باید به لایه سرویس‌های مختلف برنامه منتقل و از طریق تزریق وابستگی‌ها در دسترس باشند.
-در صفحه لاگین سایت، autocomplete خاموش شود.
- تعداد رفت و برگشت‌های به بانک اطلاعاتی در یک صفحه توسط پروفایلرها بررسی شده و اطلاعات عمومی پرمصرف کش شوند
.
- هیچکدام از کلاس‌های ASP.NET MVC نباید از HttpContext مستقیما استفاده شود.
-تمام درخواست‌های jQuery Ajax 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 پروفایلرها مطلب هست در سایت به اندازه کافی.
- در مورد مسایل 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 ماژول سشن را حذف کنم برای کار کردن با پکیج PagedList که از ViewBag استفاده می‌کنه یه مشکل بر می‌خورم در مابقی کنترولرها سشن را غیر فعال کردم ولی در این کنترولر که از ViewBag استفاده می‌کنم نمی‌توانم ... یه راهنمایی می‌خواستم. ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۲۶ ۱۱:۵۱

- ViewBag از سشن استفاده نمی‌کند.
- TempData از سشن استفاده می‌کند. امکان تهیه TempData سفارشی که از سشن استفاده نکند نیز وجود دارد. اطلاعات بیشتر

نویسنده: پریسا زاهدی
تاریخ: ۱۳۹۳/۱۰/۲۵ ۱۵:۲

اگر امکان دارد طبق توصیه ای که شما گفتین « - از هش‌های ترکیبی استفاده شود. مستقیما از MD5 یا SHA1 استفاده نشود. » لینک منابعی رو معرفی میکنین که بشه طریقه استفاده از بهترین الگوریتم‌های هش و ترکیب آنها با هم را با سمپل بررسی کرد و در کارهای امن بکار برد ؟ متشکرم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۲۵ ۱۹:۷

برای نمونه حاصل هش SHA1 منحصربفرد است. هش این هش هم با یک الگوریتم هش دیگر دلخواه (هش ترکیبی)، منحصربفرد است.

عنوان: **استفاده از FluentValidation در ASP.NET MVC**
نویسنده: محمّد زارع
تاریخ: ۱۳۹۱/۰۸/۲۰ ۱۶:۵
آدرس: www.dotnettips.info
برچسب‌ها: MVC, Validation, FluentValidation

برای هماهنگی این کتابخانه با ASP.NET MVC نیاز به نصب FluentValidation.Mvc3 یا FluentValidation.Mvc4 از طریق Nuget یا دانلود کتابخانه از سایت CodePlex می‌باشد. بعد از نصب کتابخانه، نیاز به تنظیم FluentValidationModelValidatorProvider داخل متد 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 کرده ایم که از PersonValidator جهت اعتبارسنجی استفاده کند.

در آخر می‌توانیم Controller و View ی برنامه مان 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 موجود در MVC 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 شرط‌های When/Unless استفاده کرده اند، Validatorهای سفارشی، و 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

صفت CustomizeValidator

با استفاده از CustomizeValidatorAttribute می‌توان نحوه اجرای Validator را تنظیم کرد. به عنوان مثال اگر میخواهید که Validator تنها برای یک RuleSet مخصوص انجام شود می‌توانید مانند زیر عمل کنید:

```
public ActionResult Save([CustomizeValidator(RuleSet="MyRuleset")] Customer cust) {
  // ...
}
```

مواردی که تا اینجا گفته شد برای استفاده در یک برنامه‌ی ساده MVC کافی به نظر می‌رسد، اما از اینجا به بعد مربوط به مواقعی است که نخواهیم از Attributeها استفاده کنیم و کار را به IoC بسپاریم.

استفاده از Validator Factory با استفاده از یک IoC Container

Validator Factory چیست؟ Validator Factory یک کلاس می‌باشد که وظیفه ساخت نمونه از Validatorها را بر عهده دارد. اینترفیس IValidatorFactory به صورت زیر می‌باشد:

```
public interface IValidatorFactory {
   IValidator<T> GetValidator<T>();
   IValidator GetValidator(Type type);
}
```

ساخت Validator Factory سفارشی:

برای ساخت یک Validator Factory شما می‌توانید 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 می‌کند را بررسی خواهیم کرد. ابتدا نیاز به ثبت Validatorها در

StructureMap داریم:

```
ObjectFactory.Configure(cfg => cfg.AddRegistry(new MyRegistry()));

public class MyRegistry : Registry {
    public MyRegistry() {
        For<IValidator<Person>>()
    .Singleton()
    .Use<PersonValidator>();
    }
}
```

در اینجا StructureMap را طوری تنظیم کرده ایم که از یک Registry 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. متد CreateInstance نوع مناسب را نمونه سازی می‌کند (CustomerValidator) و آن را بازمی گرداند (یا Null بر می‌گرداند اگر نوع مناسبی وجود نداشته باشد)

استفاده از AssemblyScanner

FluentValidation دارای یک AssemblyScanner می‌باشد که کار ثبت Validatorها داخل یک اسمبلی را راحت‌تر می‌سازد. با استفاده از AssemblyScanner کلاس MyRegistery ما شبیه 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 متد 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 FluentValidaion 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 حل ی وجود داره؟
بهتره این اعتبارسنجی‌ها تو کدوم لایه نوشته بشن؟

نویسنده: میثم زارع
تاریخ: ۱۳۹۱/۰۸/۲۱ ۹:۴۷

در مورد سوال اول : Custom validators with client side و Enable custom validator client side in FV
در مورد سوال دوم هم، اکثر مواقع 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

StructureMapValidatorFactory که در Application_Start() استفاده شده است، د رکجا 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

عنوان: **نحوه اجباری کردن استفاده از WWW در ASP.NET MVC**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۹/۰۱ ۲۱:۵
آدرس: www.dotnettips.info
گروه‌ها: MVC

دو آدرس www.site.com و site.com را درنظر بگیرید. در حالت متداول، هر دو به یک معنا هستند و هر دو به ریشه یک سایت اشاره می‌کنند؛ اما از دیدگاه مسایل اعتبار سنجی، خیر. کوکی‌های این دو یکسان نبوده و برای کاربران مشکل ساز خواهند شد. کاربری که از طریق آدرس site.com به سایت وارد شده، زمانیکه به لینک مفروض www.site.com وارد می‌شود (مثلا 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.RequestContext.HttpContext.Request.IsLocal)
            {
                string url =
filterContext.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 actionها پاسخ داده نمیشه، باز کماکان با مشکل قبلی مواجهیم. مثلا در رابطه با هندلرها 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 یه 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 تقریبا میشه گفت روی همه سرورها نصبه. یا اگه هم نصب نباشه، اکثر سرورا 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 -مخصوصا گوگل- با سایتهایی ـه که هر دو آدرس با و بدون wwwشون در دسترسه. یعنی مثلا از دیدگاه گوگل، حتما 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 عموما به صورت نسبی مقدار دهی می‌شوند، صرفا 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. وگرنه 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 دقیقا 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 برنامه ای با ASP.NET MVC 3 مینویسه، حداقل به یه سرور با IIS 7 دسترسی داره. ولی در حالت کلی گفته شما صحیحه. خوب در مورد HttpModule و UrlRewrite هم کاربر تنها یک بار ریدایرکت میشه. واقعا متوجه نشدم که منظورتون از «تمام آدرس‌های سایت بازنویسی شوند» چیه.
- درسته متوجهم.
چرا کاملا درسته. در کنار این روش بنده هم دو تا روش دیگه (که به نظر شخصی خودم ساده‌تر و کاراتر هستن) رو معرفی کردم. هر سه تاشون عملا یه 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 وبلاگ‌ها که فایل‌های تصاویر، js، css و غیره خود را در سایت‌های 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 تفاهم 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 cdn.site.com یا static.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 نتونستم منظورمو برسونم.
- باز هم حق با شماست. بنده هم عرض نکردم نمیشه هاست کرد. خود من هم قبلا اینکارو انجام دادم (روی سرور برتینا؛ اردیبهشت 90). ببینید، منظورم اینه که مثلا امروز اکثرمون Win 7 یا 8 رو سیستممون داریم یا بیشترمون مثلا VS 2012 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 نویس ها، فکر میکنم کمتر کسی پیدا بشه که امروز تمایل داشته باشه کلا هیچ سایتی رو (حالا MVC بودن یا نبودنش یه طرف) روی IIS 6 هاست بکنه. چند وقت پیش یادمه یه مطلبی نوشته بودید درباره چک لیست برنامه‌های MVC و اون تو یه جمله ای که خیلی خوشم اومد و تو ذهنم حک شد این مضمون بود که IE 6 و 7 رو به رحمت ایزدی بپیوندانیم D: من منظورم دقیقا در همین راستاست. بعنوان نمونه من دقیقا 2 روز پیش یه هاستی رو ثبت کردم که روی ویندوز سرور 2012 با SQLServer 2012 و دات نت 4.5 و IIS 8 با 1 گیگ فضا و پهنای باند نامحدود و 300 مگ رم، هزینه یک ساله ـش با یه دامنه com شد فقط 60 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 معتبرتر بود، گرونتر میداد که گرونش اینقدر شد. نمیدونم تونستم 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 رو حذف کنم، نمی‌دونم تا چه اندازه برای کیفیت سایت مفید هست ولی بیشتر سایت‌های که من می‌بینم این کار رو انجام می‌دند. من با asp.net form کار می‌کنم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۰۸ ۱۸:۲۵

« مروری بر قابلیت جدید ASP.NET FriendlyUrls »

« Url Routing در ASP.Net WebForms »

« UrlRewriter توسط Intelligencia.UrlRewriter »

نویسنده: بهاره فیضی
تاریخ: ۱۳۹۳/۰۷/۰۳ ۱۲:۸

سلام؛ همین کاری که فرمودید در اینجا رو چطور میشه در asp.net معمولی انجام داد؟ با توجه به اینکه IIS هم 6 هست. ممنون.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۰۳ ۱۳:۶

- نسخه‌ی بهبود یافته مطلب جاری در اینجا ارسال شده «فیلتر نهایی اصلاح شده: ForceWww.cs »
- تمام مطالب آن به سادگی توسط یک HTTP Module برای وب فرم‌ها قابل بازنویسی است. برای مثال: ForceWWWModule.cs

عنوان: **نمایش رکوردها به ترتیب اولویت به کمک jQuery UI sortable در ASP.NET MVC**
نویسنده: امیرحسین مرجانی
تاریخ: ۱۳۹۱/۰۹/۰۸ ۲۰:۴۷
آدرس: www.dotnettips.info
برچسب‌ها: ASP.Net MVC, jQuery, jQuery Ajax, UI, MVC

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 کار هایی 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.

از توضیحاتی که قبلا دادم مشخص است که این کار احتمالا 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 می کنیم) 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. Table من چنین وضعیتی دارد.

AMIRHOSSEIN-PC.Khazar - dbo.Types

Column Name	Data Type	Allow Nulls
Id	int	☐
Title	nvarchar(200)	☑
Description	text	☑
ImagePath	varchar(500)	☑
Priority	int	☑
IsActive	bit	☐
CreatedOn	datetime	☐
CreatedBy	uniqueidentifier	☐
ModifiedOn	datetime	☑
ModifiedBy	uniqueidentifier	☑
IsDeleted	bit	☐
		☐

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 فایل هایی 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 تعریف کردم برای CSS و JavaScript . و فایل‌های مربوط به Sort کردن را در صفحه ای 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>
```

فایل Index.chtml در پوشه کنترلر Type

```
@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.chtml 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>
            </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
label label-warning" })
                    |
                    @Html.ActionLink("مشاهده", "Details", new { id = item.Id }, new { @class =
"btnDetails label label-info" })
                    |
                    @Html.ActionLink("حذف", "Delete", new { id = item.Id }, new { @class = "btnDelete
label label-important" })
                </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 با خبر 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 شد تابع مربوطه اجرا می‌شود. ما در این تابع، ترتیب جدید سطرها را می‌گیریم (** به کمک مقدار Id که به هر سطر دادیم ، این مقدار Id برابر بود با Id خود رکورد در Database) و به کمک jQuery.ajax به تابع Sort از کنترلر Type در منطقه (Administrator) area ارسال می‌کنیم و در آنجا ادامه کار را انجام میدهیم.

تابع fixHelper هم به ما کمک می‌کند که هنگامی که سطرها از جای خود جدا می‌شوند ، 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 می‌شود اینگونه طراحی کردم .

```
public EmptyResult Sort(string s)
{
    if (s != null)
    {
```

s "1,2,3,4,6,5"

```
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 ابتدا مقادیر 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 اضای 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 کاملا واضح است و نیاری 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

عنوان	توضیحات	فعال	
مدل اهرمی	این مدل از طرح هاي جديد و نوين بهره مي برد.	☑	ویرایش | مشاهده | حذف
مدل کلاسیک	طرح هاي قديمي شيرآلات	☑	ویرایش | مشاهده | حذف
سینک شاوری	توضيحاتي در مورد سينک هاي شاوري	☑	ویرایش | مشاهده | حذف
دسته ها و پوسته ها	توضيحاتي در مورد دسته ها و پوسته ها	☑	ویرایش | مشاهده | حذف
علم های دوش	توضيحاتي در مورد علم هاي دوش	☑	ویرایش | مشاهده | حذف
سایر محصولات		☑	ویرایش | مشاهده | حذف

این عکس مربوط به است به قسمت مدیریت پروژه شیرآلات مرجان خزر .

نظرات خوانندگان

نویسنده: سعید
تاریخ: ۱۳۹۱/۰۹/۰۸ ۲۱:۱۲

« این کار احتمالا در ASP.NET WebForm کار سختی نیست »

اتفاقا کار ساده‌ای نیست و همین مراحل باید طی شود. ضمن اینکه کار با jquery ajax در آنجا به این یک دستی نیست. نیاز است در code behind فرم، متد وب 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 هندلر مجزا بجای یک اکشن متد کمک گرفته شود ... خلاصه خیلی داستان دارد.

نویسنده: امیرحسین مرجانی
تاریخ: ۱۳۹۱/۰۹/۰۸ ۲۱:۱۶

من خودم این کار رو در webform انجام ندادم و خیلی هم با سختی هاش آشنایی 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: ۱۳۹۱/۰۹/۰۸ ۲۱:۲۰

بسیار عالی.

فقط در مورد update panel ... تنها کاری که انجام می‌دهد کپسوله کردن ارسال مقادیر به سرور است به صورت ajax سازگار با کنترل‌های دارای view state . بیشتر از این کاری انجام نمی‌دهد. مابقی آن اگر یک سری کنترل در toolkit آن موجود باشد برای این کارها یا خیر. این toolkit هم محدود است و آنچنان به روز نمی‌شود.

نویسنده: امیرحسین مرجانی
تاریخ: ۱۳۹۱/۰۹/۰۸ ۲۱:۲۳

ممنونم 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: ۱۳۹۱/۰۹/۰۸ ۲۲:۴

سلام
مطلب مفیدی بود ولی می‌تونستید بیشتر توضیح بدید.
برای ما افراد مبتدی درک بعضی قسمت‌ها کمی مشکل هست.
ولی باز هم ممنونم

نویسنده: امیرحسین مرجانی

تاریخ: ۱۳۹۱/۰۹/۰۸ ۲۲:۷

سلام
بفرمایید کدوم قسمت‌ها تا بیشتر توضیح بدم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۹/۰۸ ۲۲:۲۱

پیشنیازهای مطلب جاری:
- در مورد RenderSection : (^)
- مقدمه‌ای بر jQuery Ajax در MVC : (^)
- db.SaveChanges : کل مباحث Entity framework سایت (^)
- EmptyResult و کلا خروجی‌های اکشن متدها: (^)
- علت استفاده از Sort@ Url.Action در حین آدرس دهی: (^)

نویسنده: امیرحسین مرجانی
تاریخ: ۱۳۹۱/۰۹/۰۸ ۲۲:۲۳

واقعا تشکر می‌کنم بابت این لیستی که ارائه کردید.
بحث تکمیل شد
به نظر شما باید بیشتر به جزییات اهمیت داده می‌شد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۹/۰۸ ۲۲:۴۱

ضرورتی نداره. چون واقعا به اندازه لیستی که عنوان شد نیاز به پیشنیاز 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: ۱۳۹۱/۰۹/۰۹ ۰:۲۴

با استفاده از Web API Controller تقریبا به همین راحتی امکانپذیره. تغییر زیادی هم نیاز نداره.

نویسنده: امیرحسین مرجانی
تاریخ: ۱۳۹۱/۰۹/۰۹ ۰:۳۷

ممنونم آقای موسوی
قابل توجه آقا سعید (احتمالا این مطلب برای شما مفیده)

نویسنده: سعید
تاریخ: ۱۳۹۱/۰۹/۰۹ ۰:۵۲

- نیاز به vs2012 داره.
- نمیشه یک فرم رو strongly typed تعریف کرد مثل مثال بالا:

```
@model IEnumerable<KhazarCo.Models.Type>
```

نویسنده: محسن موسوی
تاریخ: ۱۳۹۱/۰۹/۰۹ ۱۰:۲۱

بحث در مورد UI نبود.بحث در مورد استفاده کردن بود.
در ضمن در Web Formها نیز از امکاناتی شبیه به MVC می‌توان بهره برد.(^)

نویسنده: سعید
تاریخ: ۱۳۹۱/۰۹/۰۹ ۱۰:۴۶

فقط mvc هست که امکان استفاده از چند view engine را با هم دارد.

صفحه‌ای که لینک دادید مربوط است به web pages asp.net و نه web forms. این web pages برای کار با webmatrix طراحی شده.

نویسنده: محسن موسوی
تاریخ: ۱۳۹۱/۰۹/۰۹ ۱۱:۴

- 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: ۱۳۹۱/۰۹/۰۹ ۱۱:۴۵

```
<iframe src="/twitter" frameborder="0" height="400" width="270" scrolling="no"></iframe>
```

فلسفه و کاربرد web pages متفاوت است. در مورد وب فرم‌ها هم نمی‌تونید از razor استفاده کنید چون در یک فایل نمی‌شود از دو موتور view استفاده کرد. موتور view وب فرم‌ها، نامش همین web forms view engine است و قابل تعویض هم نیست (برخلاف MVC). در مقاله‌ای که لینک دادید، داره از یک *iframe* استفاده می‌کنه برای الحاق razor.

به علاوه در مورد Web API که کلا مطلب جدیدی نیست. نسخه ساده شده WCF است.

نویسنده: محسن موسوی
تاریخ: ۱۳۹۱/۰۹/۰۹ ۱۲:۲۷

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 WebPage و یا WebForm انجام داد؟!
چه از طریق قابلیت‌های ASP.NET Web Form و ASP.NET Web Page راحته.
استفاده از Razor چه از طریق Jquery و یا Iframe (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
یا بطور کامل از Razor View Engine در Web Page
در نهایت کار سختی نیست.

نویسنده: سعید
تاریخ: ۱۳۹۱/۰۹/۰۹ ۱۲:۴۳

razor بی‌جهت اینجا طرح شد. من در مورد strongly typed view سطری رو نوشتم. این نوع viewها مستقل از نوع view

engine هستند. در mvc حتی با موتور web forms هم میشه یک چنین viewهای تحت نظر کامپایلری رو با پشتیبانی کامل از intelisense داشت.

عنوان: **کنترل عمومی فایل‌های آپلودی در ASP.NET MVC**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۹/۰۹ ۱۳:۲
آدرس: www.dotnettips.info
گروه‌ها: MVC, Security

در مطلب « محدود کردن کاربرها به آپلود فایل‌هایی 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 کلی‌تری را درنظر بگیریم که کاربر قرار است هر نوع فایل دلخواهی را ارسال کند. در اینجا نباید امکان آپلود هر نوع فایلی، خصوصا فایل‌های اجرایی ASP.NET یا هر نوع موتور اجرایی دیگری که ممکن است روی سرور نصب باشد (مثلا PHP)، وجود داشته باشد. برای این منظور فیلتر دیگری به نام AllowUploadSafeFiles طراحی شده است که سورس آن‌را 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", ".asptxt"
        };

        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: ۱۳۹۳/۰۷/۰۲ ۸:۲

سلام
ایا می‌توان این پیام خطا را به در یک صفحه خاص نمایش داد.یا اینکه با addModelError انرا به Modelstate مربوط به یک action خاص اضافه کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۰۲ ۹:۵۹

صدور استثناء را با filterContext.Controller.ViewData.ModelState.AddModelError تعویض کنید.

عنوان: **هدایت خودکار کاربر به صفحه لاگین در حین اعمال Ajax ایی**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۹/۲۴ ۱۳:۱۷
آدرس: www.dotnettips.info
گروه‌ها: MVC, jQuery Ajax, Security

در ASP.NET MVC به کمک فیلتر Authorize می‌توان کاربران را در صورت درخواست دسترسی به کنترلر و یا اکشن متد خاصی در صورت لزوم و عدم اعتبارسنجی کامل، به صفحه لاگین 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 هدایت می‌شود. اما در مورد اعمال Ajax ایی چطور؟ در این حالت خاص، فیلتر Authorize قابلیت هدایت خودکار کاربران را به صفحه لاگین، ندارد. در ادامه نحوه رفع این نقیصه را بررسی خواهیم کرد.

تهیه فیلتر سفارشی SiteAuthorize

برای بررسی اعمال Ajaxایی، 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 اعتبار سنجی نشده از نوع Ajax ایی خاتمه داده شده و سپس StatusCode ممنوع (403) به کلاینت بازگشت داده می‌شود. در این حالت کلاینت تنها کافی است StatusCode 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 را ملاحظه می‌کنید. View ارسال کننده اطلاعات به اکشن متد 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>
}
```

تنها نکته جدید کدهای فوق، بررسی xhr.status == 403 است. اگر فیلتر SiteAuthorize کد وضعیت 403 را بازگشت دهد، به کمک مقدار دهی window.location، مرورگر را وادار خواهیم کرد تا صفحه کنترلر login را نمایش دهد. این کد جاوا اسکریپتی، با تمام مرورگرها سازگار است.

نکته تکمیلی:

در متد handleAjaxRequest، 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 ایی از سرور، تغییر می‌دهید، صفحه لاگین 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: ۱۳۹۱/۰۹/۲۴ ۱۴:۲۶

با سلام
ممنون از این مطلب واقعاً کاربردی، عالی بود.
اما موضوع زیر بسیار در محیط عملیاتی اتفاق می‌افتد:
فرض کنید کاربری در یک اداره درحال پر کردن یک فرم از برنامه ما است که چند ده فیلد داره، و طبیعیه که ممکن در این بین به دنبال کاری بره و برگرده و بقیه فرم رو پرکنه و در نهایت دکمه ثبت رو بزنه، مشکلی که پیش میاد اینه که به صفحه لاگین هدایت میشه و وقتی دوباره به اون فرم بر میگرده 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 دوباره نام 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: ۱۳۹۱/۰۹/۲۵ ۱۰:۵۸

با درود؛
اگه من بخوام 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 متد OnActionExecuting رور override کردم و یه همچین چکی رو دارم انجام میدم .
به نظرتون این کار درسته ؟
راه بهتری وجود داره ؟
شما چه روشی رو پیشنهادمیکنید ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۹/۲۵ ۱۱:۴۷

- می‌تونید 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 متد خاص از فیلتر سراسری یاد شده فقط کافی است از فیلتر جدید AllowAnonymous استفاده کنید.

نویسنده: شیرزادیان
تاریخ: ۱۳۹۱/۰۹/۲۵ ۲۳:۱۴

سلام؛
بسیار مفید و کارآمد بود. بازهم متشکرم

نویسنده: مهدی سعیدی فر
تاریخ: ۱۳۹۲/۱۱/۲۰ ۲۱:۱۲

من توی خطاهای لاگ شده توسط elmah توی سایتم خطای Server cannot set status after HTTP headers have been sent.
را در اجرای همین قسمت دریافت می‌کنم.
کار به درستی انجام میشه ولی لاگ سایت پر شده از این خطا. اشکال کار از کجای فیلتر فوق است؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۲۰ ۲۱:۳۲

این فیلتر اشکالی ندارد. احتمالا 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: ۱۳۹۲/۱۲/۲۵ ۱۳:۵۱

در حالتی که کاربر وارد شده و Authorize مقدار true دارد و ولی Role مد نظر را ندارد چطوری می‌توان کاربر را به صفحه لاگین هدایت کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۲۵ ۱۳:۵۵

اگر Role Provider تعریف 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 = (int)HttpStatusCode.Forbidden را هم می‌توانید پیش از صدور استثناء فراخوانی کنید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۱۷ ۱۴:۳۵

یک نکته‌ی تکمیلی

- برای رفع این مشکل (تداخل Forms authentication و تنظیم StatusCode) اگر از دات نت 4.5 به بعد استفاده می‌کنید، باید SuppressFormsAuthenticationRedirect را نیز پیش از ctx.Response.StatusCode اضافه کنید. به Response.End هم نیازی نخواهد بود.
- اگر از دات نت 4 استفاده می‌کنید، پیاده سازی SuppressFormsAuthenticationRedirect مخصوص آن‌را نیاز خواهید داشت.

عنوان: **مدیریت سفارشی سطوح دسترسی کاربران در MVC**
نویسنده: سید مهران موسوی
تاریخ: ۱۳۹۱/۱۰/۰۸ ۲۳:۱۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, ASP.Net MVC

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 قرار گرفته . برای شروع ما نیاز داریم یک نمونه از این کلاس رو پیاده سازی کرده و چند تابع از اون رو 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 میبینید 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 میتونید 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 واکشی لیست مجوزهای ثبت شده برای هر کاربر بررسی انجام میشه که ایا اولا کاربر یک کاربر ثبت شده هست و ثانیا اینکه ایا درخواستی که ارسال کرده 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 . مثلا 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 واکشی در کوکی ذخیره میشه و دیگه هر بار، بار اضافه برای واکشی از بانک اطلاعاتی به برنامه تحمیل نمیشه ...

حالا به این صورت میتونیم مثلا 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 ... میتونه در نظر گرفته بشه .

نظرات خوانندگان

نویسنده: امیرحسین مرجانی
تاریخ: ۱۳۹۱/۱۰/۰۹ ۰:۱۵

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 هرچند قدیمیه ولی واقعا مفیده.
به خود من که خیلی کمک کرد.

نویسنده: سعید
تاریخ: ۱۳۹۱/۱۰/۰۹ ۲:۳۱

ممنون. کلاس‌های users و roles به چه نحوی تعریف شدن؟ به نظر می‌رسه رابطه many-to-many است.

نویسنده: سید مهران موسوی
تاریخ: ۱۳۹۱/۱۰/۰۹ ۱۰:۲

سه جدول user و UserRoles و Roles در نظر گرفته شده که جدول UserRoles بین دو جدول دیگه واسط هست . هر کاربر میتونه چندین مجوز داشته باشه و هر مجوز میتونه برای چندین کاربر باشه . ارتباط n به n بین دو جدول Roles و user به واسطه‌ی جدول UserRoles برقراره.
توجه به این نکته حائز اهمیته که این مدل فقط یک مثال ساده صرفا مربوط به این مقالست و هیچ جنبه‌ی دیگه ای نداره و همونطور 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: ۱۳۹۱/۱۰/۰۹ ۱۰:۱۳

دوست عزیز RoleProvider یک اینترفیس نیست بلکه یک کلاس مشتق شده از ProviderBase هستش که اونم از کلاس system.object مشتق شده . برای اطلاعات بیشتر توجهتون رو به صفحه‌ی مربوطه در msdn جلب میکنم : RoleProvider

نویسنده: علیرضا
تاریخ: ۱۳۹۱/۱۰/۰۹ ۱۱:۱۵

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 ... میتونه در نظر گرفته بشه .]

نویسنده: سید مهران موسوی
تاریخ: ۱۳۹۱/۱۰/۰۹ ۱۲:۱۹

شما فیلتر Authorize رو میتونید 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 تیمم با استفاده از Reflection، اسامی متدهایی که خروجی ActionResult دارند رو بازیابی می‌کنیم و در سیستم امنیت و پایه برای مدیر امنیت و برنامه نویس نشان میدیم و اونها می‌تونن دسترسی رو بر این مبنا تنظیم کنن. وجود یک Contoller پایه و یک فیلتر برای اون با override کردن متد OnActionExecuting از الزامات کار هست.

نویسنده: سید مهران موسوی
تاریخ: ۱۳۹۱/۱۰/۱۰ ۱۲:۷

جالبه . من در پروژه ای 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 اکش roleهای دلخواه رو انتصاب 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 CashManagment هم قرار داره که از بار اظافه‌ی 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 . اونها 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 برخورد 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 کدوم هم 10 مجوز براشون 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 بدبینانه‌ترین 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، اِعمال بلافاصله‌ی 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 Cache 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 واکشیش با سرعت بسیار بالا انجام میشه بررسی میشه و اگه نام اون کاربر در اون لیست بود مجددا میتونیم 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 مربوطه 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

مرسی از توضیحاتت.

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 Connection اضافه کل سیستم زیر سوال بره، بهتره بساط اون پایگاه داده جمع بشه.
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 AJAXیی مجوزش گرفته شد چه واکنشی باید رخ بده؟ 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 حالت ها) و ایجاد ارتباط ساده با پایگاه داده برای بررسی داشتن مجوز کاربر انجام بده.

نویسنده: سید مهران موسوی
تاریخ: ۱۳۹۱/۱۰/۱۰ ۲۱:۴۲

اقای راد من هیچوقت قصد شعار دادن بر مبنای تحلیلات ذهنیم رو نداشتم و نخواهم داشت .
تمام حالت هایی 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 کدوم از پروژه هایی که لازم باشه به سادگی ممکن میشه (به نظرم برنامه نویسی شیء گرا زیباست) . اگر توسعه دهندگان همون پایگاه‌های داده‌ی 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 نمیشدند ... 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 واقعا کار کردن باهاشون عذاب آور و کسل کننده باشه...

انشاالله در چندین ماه اینده بعد از کامل شدن سیستم و انتشار اون میتونید 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 نداشتم به شما و دیگر دوستان . این سایت محیط مقدسی هست چون واقعا مطالب خوبی در اون قرار میگیره امیدوارم بتونم از اطلاعات شما و دیگر دوستان استفاده کنم . یا حق

نویسنده: ابوالفضل
تاریخ: ۱۳۹۱/۱۰/۱۲ ۱۳:۴

سلام
من از سیستم معمولی وب فرم برای تشخیص هویت در mvc 4 استفاده کردم. اما به یه مشکل جالب برخوردم.
اپلیکیشین 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 همیشه false برمیگردونه و نمی‌تونم وارد شم. با هیچکدوم از مرورگرها وارد نمیشه ولی با **فایرفاکس** میتونم وارد شم!
من بدین صورت کد لوگین رو نوشتم:

```
if (Membership.ValidateUser(UserName, Password))
            {
                FormsAuthentication.RedirectFromLoginPage(UserName, true);
                return true;
            }
```

دوستان نظرشون چیه؟ مشکل از چی می‌تونه باشه؟ فایرفاکس چه خاصیتی داره که میتونه 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 18 سری MVC به متد public ActionResult LogOn ، یک FormsAuthentication.SetAuthCookie هم باید پیش از Redirect اضافه شود.

نویسنده: ابوالفضل
تاریخ: ۱۳۹۱/۱۰/۱۲ ۱۴:۴۶

با تشکر آقای نصیری. حل شد. مشکل همون بود. کلی سرچ 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 می‌کردم. الان فهمیدم قضیه این بوده! :)
ولی باز هم برام سوال هست که چرا فایرفاکس لوگین میکرد ولی بقیه نه؟!

نویسنده: مهتدی حسن‌پور
تاریخ: ۱۳۹۱/۱۱/۱۴ ۱۲:۲۵

با سلام جناب راد
امکانش هست کمی بیشتر درمورد 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.
- IgnoreCase 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 واکشی 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 کوکی‌های مروگر چیزی نمی‌بینم، در ضمن یک Break Point هم سر متد GetRolesForUserگذاشتم که هر وقت قصد دیدن یک اکشن برای یک Role رو مشخص دارم این متد اجرا میشه یعنی از کوکی نقش رو نمی‌خونه.
سوال بعدیم اینه که این نقش چطوری توی مرورگر کاربر ذخیره میشه؟ آیا رمز گزاری شده هست یا نه به صورت Plain Text ذخیره میشه؟ اگه رمزگزاری 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 شده‌ای 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 IsInRole یا GetRoles تولید می‌شود و نه پیش از آن.
- اگر از دات نت 4 و نیم استفاده می‌کنید، برای حالت کش نشدن این نقش‌ها اخیرا 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: ۱۳۹۲/۰۱/۰۶ ۱:۴

سلام

اگر کاربر بتونه متن Role رو ببینه و همچنین تغییرش بده... آیا این مشکل پیش نمیاد که شخصی بیاد و متن کوکی مدیریت رو

برداره و زمانی که لوگین کرد متن کوکی خودش رو تغییر بده. اینجوری چون همه رول‌ها از کوکی خونده میشه پس می‌تونه به این شیوه به رول‌ها کلک بزنه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۱/۰۶ ۱:۶

این اطلاعات رمزنگاری 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 نوشتید 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، فعلا 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 Gtihub فعلا متاسفانه ندارم.
فعلا بازده عالی ای داشته این CMF از همه جهات ، اگر تصمیم به انتشار عمومیش گرفتم حتما 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 رو از RoleProvider ارث بری میکنم، به جز دو تابع GetRolesForUser و IsUserInRole که override کردیم 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 18 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 متد GetUserById را صدازد و بار بر روی سیستم را افزایش داد آیا روش بنده صحیح است از لحاظ performance و امنیت؟

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۱۶ ۱۴:۳۴

اتفاقا 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 رمزنگاری شده در مرورگر کاربر ذخیره خواهد شد. در این حالت مصرف حافظه کمتری را در سمت سرور خواهید داشت و ضمنا با ری‌سایکل شدن برنامه در IIS، تمام لاگین‌های کاربران از دست نخواهد رفت (اصطلاحا تمام سشن‌های لاگین نمی‌پرند) و مجبور به لاگین مجدد نخواهند شد؛ چون کوکی رمزنگاری 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 ببینیم کاربری نمایندست 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 رمزنگاری شده آن قرار می‌گیره. یعنی واکشی از دیتابیس نداره. ضمنا RoleProvider استفاده شده هم زمانیکه در وب کانفیگ تنظیم میشه، مقدار cacherolesincookie=true داره (در مقاله هست). یعنی این هم کش میشه و هربار از دیتابیس واکشی نمیشه.

نویسنده: وحید
تاریخ: ۱۳۹۲/۱۰/۱۶ ۲۳:۲۷

ممنون از شما آیا با cacherolesincookie=true فقط roleها کش میشوند ؟ مثلا زمانی که کاربری لاگین کرده و در هدر پیغام خوش آمدید کاربر گرامی آقای...می شود این نام و نام خانوادگی را باید هر دفعه بازای هر request از دیتابیس بیاریم یا میتونیم از حالت شبیه این cacherolesincookie=true استفاده کرد با تشکر

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۱۶ ۲۳:۳۸

برای این حالت‌های خاص می‌تونید از CacheManager قسمت 19 سری MVC 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 Mvc 18 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 می‌کردیم که از دید کاربر به دور بود ، اما برای پردازش موازی و انجام چک لیست تهیه شده بهتره که از Sessionها استفاده نشه،
خب حالا ذخیره این اطلاعات در کوکی‌ها درسته؟ حتی اگر کد بشن باز خطرناکه ، چرا که یک کاربر اگر بتونه کوکی که مدیر بودن یا نبودن کاربر را مشخص می‌کند را تغییر بده می‌تونه به همه بخش‌ها دسترسی پیدا کنه!
از طرفی صرفا مدیر بودن یا نبودن یک کاربر مطرح نیست ، اطلاعات زیادی هستند که مهم هستند و ذخیره اون‌ها در Session میتونه منابع سرور و به خودش مشغول کنه!
روش درست چیه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۲۰ ۹:۲۸

- راه حل‌های مبتنی بر سشن، از Classics ASP دهه نود به ارث رسیده‌اند. عملا با پیشرفت‌هایی که حاصل شده نیازی به بسیاری از آن‌ها نیست. مصرف حافظه بالایی دارند و همچنین با ری‌استارت شدن برنامه در سرور، تمام سشن‌ها از بین خواهند رفت. این مشکلات در Forms Authentication وجود ندارند.
- قدمت Forms Authentication به ASP.NET *1.x* بر می‌گردد. می‌توانید در این مورد در سایت‌های دیگر نیز بیشتر تحقیق کنید که آیا مشکل حادی از سال 2001 تا الان گزارش شده یا خیر.
- کلید رمزنگاری این کوکی‌ها در سمت سرور قرار دارد و تنها یک راه برای دسترسی به آن‌ها 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 کوکی‌های آن قابل تنظیم است (بحث شده در مطلب فوق).

نویسنده: رضا گرمارودی
تاریخ: ۱۳۹۲/۱۱/۲۰ ۱۰:۴۵

ممنون از پاسختون اما Authenticate در مند SetAuthCookie تنها نام یوزر و ذخیره می‌کند، برای ذخیره سایر اطلاعات چه کار باید کرد؟

اگر باید از متدهای HttpContext استفاده بشود جهت رمزنگاری کوکی‌ها چه الگوریتمی به اندازه Authentication مطمئن است؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۲۰ ۱۱:۲۸

- سایر اطلاعات مانند نقش‌های یک کاربر (مدیر است یا نویسنده مثلا) در مطلب جاری به صورت یک role provider مکمل پیاده سازی شده‌است.
+ بیشتر از این نیازی نیست اطلاعاتی را در کوکی‌ها یا جای دیگری ذخیره کنید. اطلاعاتی فراتر از این (مانند صفحه پروفایل یک شخص در سایت)، بسیار شخصی بوده و هر زمانیکه نیاز باشد، باید مستقلا از دیتابیس واکشی شوند.

نویسنده: مهدی سعیدی فر
تاریخ: ۱۳۹۲/۱۱/۲۰ ۱۳:۴۳

بنده در حال توسعه‌ی یه 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 می‌گذارم.

نویسنده: رضا گرمارودی
تاریخ: ۱۳۹۲/۱۱/۲۳ ۱۲:۱۸

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 2008و7وvista 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 درصورت وجود کش اطلاعات می‌خونه. اصلا چرا در دات نت 4.5 این مشکل هست و چرا بر روی ویندوز 8 این پیغام داده میشود؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۲۳ ۱۲:۴۴

- مشکل طولانی بودن حاصل BinaryFormatter serialization برطرف شده (نزدیک 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 ویندوز 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.

نویسنده: ارکا
تاریخ: ۱۳۹۲/۱۲/۱۵ ۲۱:۳۷

سلام،
الان دو، سه روزه که میخوام شخصی سازی Membership رو یاد بگیرم، اما واقعا 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 داره؟
- من Web Form کار میکنم.
ممنون

نویسنده: محمد شهریاری
تاریخ: ۱۳۹۲/۱۲/۱۶ ۸:۳۰

مبحث مربوط به مدیریت کاربران و گروه کاربران و پروفایل از طریق سه کلاس پایه System.Web.Security.MembershipProvider
System.Web.Security.RoleProvider
System.Web.Profile.ProfileProvider
همونطور که از کلاس‌ها مشخص هست به صورت Provider مدیریت می‌شود. این به این معنی است که اگر شما بخواید Provider خودتون با هر Datasource جدید یا منطق جدید داشته باشید مجاز هستید نسخه جدید از این Providerها رو 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 ODBC , EF , .. 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: ۱۳۹۲/۱۲/۱۶ ۹:۲۸

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: ۱۳۹۲/۱۲/۱۶ ۱۰:۵۳

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 سؤالات 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: ۱۳۹۳/۰۵/۱۰ ۱۲:۴۲

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 بزازین ممنون میشم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۱۰ ۱۲:۴۶

- 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: ۱۳۹۳/۰۵/۱۰ ۲۰:۵۷

ممنون. لینک دوم من نتونستم روی لوکال نصب کنم اما Iris membership دیدم کمی درک کردم اما نمیدونم database چطور به پروژه Add کنم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۱۰ ۲۳:۱۶

پیشنیاز درک این پروژه‌ها EF Code first است.

نویسنده: فایز مهربانی
تاریخ: ۱۳۹۳/۰۷/۲۶ ۱۳:۴۲

سلام وقت بخیر...
ببخشید در مورد Claim Authorization چه مطالبی وجود داره ؟ رفرنس فارسی وجود داره ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۲۶ ۱۴:۴۴

« مروری بر Claim »

عنوان: **بهبود SEO در ASP.NET MVC**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۰/۱۵ ۱۸:۱۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, SEO

گوگل خلاصه نتایج Indexing یک سایت را توسط ابزاری به نام Google webmaster tools در اختیار علاقمندان قرار می‌دهد. Bing نیز چنین ابزاری را تدارک دیده است.
به آمارهای خطای حاصل از سایت جاری که دقت می‌کردم یک نکته آن جالب بود: «محتوای تکراری»

```
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 نهایتا 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 (isLocalRequet(filterContext))
                return absoluteUrlToLower;

            if (absoluteUrlToLower.Contains("www"))
                return absoluteUrlToLower;

            return absoluteUrlToLower.Replace("http://", "http://www.")
                                     .Replace("https://", "https://www.");
        }

        private static bool isLocalRequet(ActionExecutingContext filterContext)
        {
            return filterContext.RequestContext.HttpContext.Request.IsLocal;
        }
    }
}
```

کلاس فوق، نگارش تکمیل شده ForceWww که پیشتر در این سایت دیده‌اید. توسط آن سه بررسی مختلف بر روی لینک جاری در حال پردازش صورت خواهد گرفت:

- تمام آدرس‌های سایت باید www داشته باشند؛ تا آدرس‌های آن یکنواخت شده و خصوصا 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 خصوصا 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 Urlهایی که در بدو امر lowercase هستند

برای پیاده سازی این مطلب می‌توان از پروژه سورس باز « LowercaseRoutesMVC » استفاده کرد. سه فایل cs دارد که می‌توانید به پروژه خود اضافه کنید. پس از آن، هرجایی در پروژه خود routes.MapRoute دارید تبدیل کنید به routes.MapRouteLowercase . به این ترتیب به صورت خودکار تمام Urlهای تولید شده توسط HTML helpers توکار ASP.NET MVC (و نه Urlهایی که دستی نوشته شده‌اند)، در حین درج در صفحه به صورت lowercase ظاهر خواهند شد (صرفنظر از اینکه نام‌های 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 کمکی نیز از آدرس‌هایی که دارای کوئری استرینگ باشند، صرفنظر می‌شود.

نظرات خوانندگان

نویسنده: امیرحسین مرجانی
تاریخ: ۱۳۹۱/۱۰/۱۵ ۲۳:۲۲

سلام
ممنونم ، مطلب مفیدی بود.
اگر ممکنه بیشتر روی بحث SEO مطلب منتشر کنید.

نویسنده: یاسر مرادی
تاریخ: ۱۳۹۱/۱۰/۱۶ ۱۰:۳۲

بهتر نبود برای تشخیص این که Request دارای Query String است یا خیر، قسمت Query String آن را بررسی می‌کردید که مقدار دارد یا خیر ؟
تا این که در کل کوئری دنبال علامت ؟ بگردید ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۰/۱۶ ۱۱:۱

- تشخیص وجود کوئری استرینگ در یک Url، حالت دیگری نمی‌تونه داشته باشه.
- ضمنا یک Url می‌تونه 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: ۱۳۹۱/۱۰/۱۶ ۲۲:۳۷

این پاسخ در stackoverflow.com به نقل از اسکات گو هم بد نیست: http://stackoverflow.com/a/170955/538387

نویسنده: محسن
تاریخ: ۱۳۹۱/۱۰/۱۷ ۰:۳۳

راه مناسبی نیست. چون قید (IRouteConstraint) اعمال می‌کنه و سبب می‌شه که خیلی از urlها در سایت پردازش نشن.

نویسنده: جلال
تاریخ: ۱۳۹۱/۱۰/۱۸ ۲:۵۵

سلام
در تابع 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، Controller یا Action دارای www باشه، مشکلی پیش نمیاد. البته نکته دیگه درباره آدرس‌های دارای query string این که چرا تا قبل از ? رو به حروف کوچک تبدیل نکنیم؟ دلیل خاصی داره؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۰/۱۸ ۸:۴۷

- این یک کار سورس باز هست. مطابق نیاز خودتون تغییرش بدید و استفاده کنید. برای استفاده‌ای 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. خصوصا قسمت کوئری استرینگ آن‌ها نباید تغییری کند چون در رمزگشایی اطلاعات از آن استفاده می‌شود.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۰/۲۶ ۱۲:۵۵

تکمیل 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=وحید نصیری" rel="nofollow">ارسال پیام خصوصی</a>
```

اگر در سایت لینک‌هایی دارید که نیاز به اعتبارسنجی و لاگین 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 ASP.NET MVC از Lowercase generated URLs :

```
public static void RegisterRoutes(RouteCollection routes)
{
    ...
    routes.LowercaseUrls = true;
    ...
}
```

نویسنده: vahid
تاریخ: ۱۳۹۲/۱۱/۰۶ ۱۳:۲۷

سلام
من با دات نت 4.5 نتونستم 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. ضمنا این تنظیم، بدون تعریف

canonical urls 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: ۱۳۹۲/۱۱/۰۶ ۱۴:۱۸

نکته‌ای مهم برای کاهش SEO جماعت کپی پیست کار

اگر محتوای سایت شما عینا 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 Contributor to).

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۱۱/۱۹ ۰:۲۱

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 نمیشه اما وقتی کد خروجی سایت بررسی می‌کنیم می‌بینم مثل بالا قسمت cononical اعمال شده، مشکل ممکنه از کجا باشه؟ در حالی که کد درست اجرا میشه اما ریدایرکت رخ نمی‌ده - این بحث چطور بررسی کنم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۹ ۰:۳۸

- روش دوم اجبار به www با استفاده از ماژول rewrite در IISهای 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 Urlهای 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>
```

ماخذ: IIS Server through the eyes of an SEO

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/|umbraco_client/|base/|webresource\.axd|scriptr
esource\.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="http://{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 ۱۳:۱۶

یک نکته‌ی تکمیلی در مورد محتوای تکراری

به گزارش‌های گوگل در مورد سایت جاری که نگاه می‌کردم، تمام لینک‌های به صفحات جستجوی سایت را مثلا

http://www.dotnettips.info/search?term=seo 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 Urlهای 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 دقیقا 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/"));
```

نویسنده: محمّد زارع
تاریخ: ۱۳۹۳/۱۰/۲۸ ۱۸:۳۵

سلام. من وقتی از این کلاس ForceWww استفاده میکنم بعد از پابلیش خطای "This webpage has a redirect loop" رو میده. میشه لطفاً راهنمایی کنید اگر نکته خاصی وجود داره

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۲۸ ۱۹:۵

- از این فیلتر در حال حاضر در همین سایت جاری استفاده می‌شود. (بنابراین مشکلی ندارد)
- قسمت new RedirectResult آن‌را در سایت خودتان لاگ کنید، تا مشخص شود چندبار و به چه علتی اجرا می‌شود. همچنین بررسی کنید آیا جای دیگری در برنامه Redirect دیگری تنظیم شده یا خیر.

عنوان: **ایجاد قسمت‌های Toggle در سایت با jQuery**
نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۱/۱۰/۲۷ ۲۱:۳۵
آدرس: www.dotnettips.info
برچسب‌ها: MVC, jQuery, ASP.Net 4.5, Design

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>کادر لغزان با jQuery</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 />
                    محتویات دلخواه خود را در این قسمت قرار دهید
                </div>
            </div>
            <div><a href="#" id="closeLogin"></a></div>
```

```
        </div>
        <div id="container">
            <div id="top">
                <!-- login -->
                <ul>
                    <li> </li>
                    <li><a id="toggleLogin" href="#">پانل باز شو</a></li>
                </ul>
                <!-- / login -->
            </div>
            <!-- / top -->

        </div>
    </div>
    <div id="main">
        محتویات 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، نگهدارنده کل قسمت مربوطه (کادر لغزان می‌باشد)، و قسمت 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 شمایل 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;
```

```
}
#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: ۱۳۹۱/۱۲/۲۱ ۲۰:۲۸

سلام آقای فتح الله ممنون از پستی که قرار دادید بنده کد شما رو تویه 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: ۱۳۹۱/۱۲/۲۱ ۲۳:۵۵

مراجعه کنید به مطلب نمایش پیغامی به کاربر در هنگام استفاده از MS Ajax و همچنین استفاده‌ی همزمان از آپدیت پنل ASP.Net و پلاگین‌های جی‌کوئری

عنوان: **آشنایی و بررسی ابزار MiniProfiler**
نویسنده: مجتبی کاویانی
تاریخ: ۱۳۹۱/۱۱/۱۵ ۱۶:۰
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, Entity framework, ASP.Net MVC, EF Code First, Oracle, Statistics, stackoverflow, MVC

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 می‌نمایید
امکان مشاهده و بررسی کوئری‌های خام ADO.NET از قبیل SQL Server,Oracle و LINQ-to-SQL و EF/First Code و...
نمایش زمان اجرای عملی صفحات

برای استفاده از این ابزار کافیست تا آن را از nuget دریافت نمایید

```
PM> Install-Package MiniProfiler
```

در ASP.NET MVC در صفحه _Layout قبل از بسته شدن تگ 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

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

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 Entity Framework/First Code کتابخانه MiniProfiler.EF را اضافه نمایید

```
PM> Install-Package MiniProfiler.EF
```

اگر بصورت دستی MiniProfiler را پیکربندی کرده باشید می‌بایست در Application_Start دستور زیر را اجرا نمایید

```
protected void Application_BeginRequest()
{
    if (Request.IsLocal)
    {
        MiniProfiler.Start();
        MiniProfilerEF.Initialize();
    }
}
```

در حالت پیشرفته‌تر اگر قصد داشته باشید زمان یک قطعه کد را جداگانه محاسبه نمایید بصورت زیر عمل نمایید

```
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 دقیقا 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 رو تست کردم با ef5 کار نمی‌کرد و خطا می‌داد. 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 ef code first 5 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 دانت ای اس پی کار نمیکنه فقط تو ام وی سی جواب میده
MiniProfiler نمیشناسه

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 خطا

```
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
```

میدهد
از mvc4 , entity framework6 استفاده می‌کنیم
در صورت امکان روش رفع خطا می‌دانید بیان نماید تا بتوانم از این ابزار استفاده کنم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۲۷ ۱۸:۲۲

برای EF 6 یک بسته‌ی نیوگت جداگانه دارد. اطلاعات بیشتر

عنوان: استفاده از Flash Uploader در ASP.NET MVC
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۲۰ ۲۲:۵
آدرس: www.dotnettips.info
برچسب‌ها: MVC, upload

چندسال قبل یک کنترل آپلود فایل در برنامه‌های ASP.NET Web forms در سایت Code projects 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 فلش نهایی هستید به پوشه OriginalFlashSource پروژه پیوست شده مراجعه کنید. در اینجا برای مثال یک سری از برچسب‌های آن فارسی شده‌اند و کامپایل مجدد.

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 دقیقا 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 درنظر 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 FlashUploadHelper.cshtml در پوشه App_Code 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 متدی است که قرار است اطلاعات فایل‌ها را دریافت کند. queryParameters اختیاری است. اگر تعریف شود تعدادی کوئری استرینگ دلخواه را می‌تواند به متد Uploader ارسال کند. برای نمونه در اینجا User و Id ارسال شده‌اند یا هر نوع کوئری استرینگ دیگری که مدنظر است.

flashUrl مسیر فایل SWF را مشخص می‌کند. در اینجا فایل FlashFileUpload.swfدر پوشه Content/FlashUpload قرار گرفته است.

fileTypeDescription برچسبی است که نوع فایل‌های قابل انتخاب را به کاربر نمایش می‌دهد و fileTypes نوع‌های مجاز قابل ارسال را دقیقا مشخص می‌کند.

پارامترهای uploadFileSizeLimit و totalUploadSizeLimit 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.locationرا مقدار دهی نمائید.

در متد Uploader کنترلر فوق، پارامترهای User و id اختیاری بوده و بر اساس queryParameters متد

FlashUploadHelper.AddFlashUploader مشخص می‌شوند. اما نام FileData نباید تغییری کند؛ از این لحاظ که دقیقا همین نام در فایل فلش، مورد استفاده قرار گرفته است.

در اکشن متد دریافت فایل‌ها، لیستی از فایل‌های 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 ویژه، به کمک امکانات Razor» در قسمت هشتم سری MVC برای مشاهده توضیحات 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-xها 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 درصد 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 کلاینتها عمل می‌کند و به صورت خودکار از نوعی که خود Browser کلاینتها می‌توانند استفاده کنند استفاده می‌نماید برای توضیحات 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 »

عنوان: **استایل دهی به ستون های header در WebGrid**
نویسنده: محسن.د
تاریخ: ۱۳۹۱/۱۱/۲۸ ۰:۲۵
آدرس: www.dotnettips.info
برچسب‌ها: ASP.Net MVC, MVC

Webgrid گرید توکار asp.net mvc 3 است که در سری آموزش‌های mvc جناب نصیری به خوبی بررسی شده است . WebGrid از طریق مجموعه ای از خواص امکان استایل دهی به ستون‌ها و ردیف‌ها را به توسعه دهنده می‌دهد . اما در این بخش مشکلی وجود دارد که در ادامه به آن خواهم پرداخت . کدهای زیر را در نظر بگیرید

مدل‌ها :

```
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);
        }
    }
```

تابع کمکی برای ایجاد گرید :

```
@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 می‌رویم

List

Id	Name	Email	Website	Phone
1	mohsen.d	email@domain.com	domain.com	213214321

خوب چندان بد نیست . با استفاده از استایل‌های تعریف شده برای فیلدها و ردیف‌ها ، لیست ساختار مناسبی دارد . اما حالا به Home/EmptyList می رویم :

List

Id	Name	Email	Website	Phone

همانطور که می‌بینید استایل هایی که برای هر ستون تعریف کرده بودیم اعمال نشده اند. مشکل هم همین جاست . WebGrid استایل تعریف شده را تنها به ستون‌های درون tbody اعمال میکند و thead از این تنظیمات بی نصیب می‌ماند (WebGrid از table برای ساختن لیست استفاده می‌کند) و در زمانی که رکوردی وجود نداشته باشد فرمت طراحی شده اعمال نمی‌شود .

```
<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 ، استایل‌های تعریف شده را در یک فیلد hidden ذخیره و سپس با استفاده از jquery این استایل‌ها را به ستون‌های header اعمال می‌کنیم .

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 می‌کنیم :

```
@SetHeaderColumnsStyle(columns)
```

و در view کد javascript زیر را اضافه می‌کنیم :

```
<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 گرید 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>
                <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 .

عنوان: **ELMAH و حملات XSS**
نویسنده: داود زینی
تاریخ: ۱۳۹۱/۱۱/۲۸ ۱۳:۴۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, Security, ASP.Net MVC, ELMAH, xss

ASP.NET 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. لاگ کردن و بررسی این خطاها جهت آگاهی از وجود حمله بی اهمیت نیست. اما متأسفانه ELMAH که به عنوان معمول‌ترین ابزار ثبت خطاها 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 خطاها به شکل زیر است (ASP.NET MVC):

```
public class ElmahRequestValidationErrorFilter : 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 ElmahRequestValidationErrorFilter());
    filters.Add(new HandleErrorAttribute());
}
```

به این ترتیب HttpRequestValidationException هم بعد از این در سیستم ELMAH ثبت خواهد شد.

نظرات خوانندگان

نویسنده: یاسر مرادی
تاریخ: ۱۳۹۱/۱۱/۲۸ ۱۴:۳

با سلام، در چک لیست ASP.NET MVC مورد زیر وجود دارد
آیا مورد زیر کماکا معتبر است ؟
- فیلتر پیش فرض مدیریت خطاها حذف و بجای آن از ELMAH استفاده شود.

با سپاس

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۱/۲۸ ۱۴:۲۶

- به سطر HandleErrorAttribute پیش فرض نیازی نیست (البته اگر تنظیمات وب کانفیگ درستی داشته باشید). در قسمت 16 سری MVC توضیح دادم. وجود آن سبب می‌شود که ELMAH اصلا کار نکند و خطای مدیریت نشده‌ای به آن ارجاع داده نشود (چون قبلا مدیریت شده).
- بله. همچنان ELMAH معتبر است. نکته فوق را هم اضافه کنید، کاملتر خواهد شد.

نویسنده: داود زینی
تاریخ: ۱۳۹۱/۱۱/۲۸ ۱۴:۲۸

سلام
این چک لیست توسط آقای نصیری تهیه شده بود. این مورد هم از نظر من معتبر است.
با تشکر

عنوان: آموزش MEF#2(استفاده از MEF در Asp.Net MVC)
نویسنده: مسعود پاکدل
تاریخ: ۱۳۹۱/۱۱/۲۹ ۲۳:۳۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, MEF, ASP.Net MVC 4

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 رو 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 فولدر 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 می‌رسه که باید به صورت زیر ایجاد بشه.

```
[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() );
        }
    }
```

PartCreationPolicyکه شامل 3 نوع می‌باشد.
Shared: بعنی در نهایت فقط یک نمونه از این کلاس در هز Container وجود دارد.
NonShared : یعنی به ازای هر درخواستی که از نمونه‌ی Export شده می‌شود یک نمونه جدید ساخته می‌شود.
Any : هر 2 حالت فوق Support می‌شود.

حالا قصد داریم یک ControllerFactory با استفاده از MEF ایجاد کنیم.(Controller Factory برای ایجاد نمونه ای از کلاس Controller مورد نظر استفاده می‌شود) برای بیشتر پروژه‌ها استفاده از DefaultControllerFactory کاملا مناسبه.

```
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 می‌تونید پست قبلی رو مطالعه کنید.
حالا قصد داریم یک DependencyResolver رو با استفاده از MEF به صورت زیر ایجاد کنیم.(DependencyResolver برای ایجاد نمونه ای از کلاس مورد نظر برای کلاس هایی است که به یکدیگر نیاز دارند و برای ارتباط بین آن از Depedency 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 CompositionContainer ایجاد کردیم که یک ComposablePartCatalog از نوع DirectoryCatalog به اون پاس دادم.

DirectoryCatalog یک مسیر رو دریافت کرده و Assemblyهای موجود در مسیر مورد نظر رو به عنوان Catalog در Container اضافه میکنه. می‌تونستید از یک AssemblyCatalog هم به صورت زیر استفاده کنید.

```
var container = new CompositionContainer( new AssemblyCatalog( Assembly.GetExecutingAssembly() ) );
```

در تکه کد زیر ControllerFactory پروژه رو از نوع MEFControllerFactory قرار دادیم.

```
ControllerBuilder.Current.SetControllerFactory( new MEFControllerFactory( container ) );
```

و در تکه کد زیر هم DependencyResolver پروژه 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 رو 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.پاکدل
تاریخ: ۱۳۹۱/۱۲/۱۹ ۸:۲۶

با توجه به این که پیاده سازی الگوی UnitOfWork در ORMهای مختلف متفاوت است یک مثال کلی در این زمینه پیاه سازی می‌کنم.
فرض کنیم یک اینترفیس به صورت زیر داریم:

```
public interface IUnitOfWork
    {
        ISession CurrentSession { get; }
        void BeginTransaction();
        void Commit();
        void RollBack();
    }
```

می‌تونید 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 Respository
    {

       [Import]

       private IUnitOfWork uow;

        public Respository()
        {
        }
    }
```

در کلاس Repository فیلد uow به دلیل داشتن Import Attribute همیشه توسط Composition Container

مقدار دهی می‌شه.

نویسنده: سینا نادی حق
تاریخ: ۱۳۹۲/۰۹/۲۰ ۱۵:۳

سلام؛ من MEF رو تو MVC اعمال کردم الان به یک مشکلی بر خوردم. ممنون میشم اگه کمکم کنید.
من Pluginها رو توی پروژه جدا با Class Library می‌نویسم، پروژه MVC هم razor engine هست. ولی وقتی می‌خوام viewها رو به Modelها نسبت بدم نمیشه(Model توی همون پروژه Class Library پلاگین هست)
کد view:

```
@model Plugin1.Models.Post
@{
    Layout = "~/Views/Shared/_Layout.cshtml";
}

@Model.Title
```

ارورشم اینه:

The type or namespace name 'Plugin1' could not be found (are you missing a using directive or an assembly reference?)

در کل چطوری می‌تونم Modelها رو به Viewها در این حالت نسبت بدم.
ممنون

نویسنده: مسعود پاکدل
تاریخ: ۱۳۹۲/۰۹/۲۰ ۱۷:۱۳

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 مدل‌های 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 DomainModel 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 حتما اسمبلی‌های مربوطه را از طریق کاتالوگ‌های موجود (نظیر AssembleyCatalog و DirectoryCatalog و...) به CompositionContainer پروژه MVC خود اضافه کنید.

نویسنده: سینوس
تاریخ: ۱۳۹۲/۱۱/۰۵ ۲۳:۴۰

با سلام و خسته نباشید. من پروژم رو با MEF راه اندازی کردم مشکل خاصی هم ندارم

با توجه به فرمایش شما باید Modelها توی یک پروژه جدا و عمومی استفاده بشه. مشکلی که من دارم اینه که وقتی پروژه (core) آماده شد و آپلود کردم دیگه نمی‌خوام وقتی کاربری از یک ماژول استفاده می‌کند خود پروژه یا پروژه مدل‌ها update بشه. دلیلشم این هست که هر کاربری ماژول‌های جداگانه دارد و ماژول‌ها رو runtime اضافه، حذف و یا بروز می‌کند.
ممنون از لطفتون

عنوان: نحوه استفاده از ViewModel در ASP.NET MVC
نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۱/۱۲/۱۳ ۱۴:۲۵
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net MVC 4, MVC

یک Model چیست؟

· قسمتی از Application است که Domain Logic را پیاده سازی می‌کند.

· همچنین با عنوان Business Logic نیز شناخته می‌شود.

· Domain Logic داده‌هایی را که بین UI و دیتابیس پاس داده می‌شود، مدیریت می‌کند.

· برای مثال، در یک سیستم انبار، Model کارش ذخیره سازی اقلام در حافظه و Logic تعیین کننده موجود بودن یک آیتم در انبار میباشد.

یک ViewModel چیست؟

· ViewModel به ما این امکان را میدهد تا از چندین Entity ، یک شیء واحد بسازیم.

· بهینه 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 Validate کردن ViewModel را به روشی متفاوت‌تر از Domain Model برای سناریوهای اعتبارسنجی attribute-based را در اختیارمان قرار میدهد.
• برای قالبندی داده‌ها نیز استفاده میشود.

o برای مثال به یک داده یا مقدار پولی قالبندی شده به یک روش خاص نیاز داریم؟
ViewModel 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 ViewModelها

داده‌هایی را که قرار است در View رندر شوند، داخل ViewModel قرار دهید.
View باید خصوصیاتی را که نیاز دارد، مستقیما از ViewModel بدست بیاورد.
وقتی که ViewModel شما پیچیده‌تر می‌شود، بهتر است از یک Mapper استفاده کنید.

اجازه دهید مثالی را در این رابطه براتون 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. ViewModel مربوطه به صورت زیر تعریف شده

است :

```
public class SpeakerAndNewsViewModel
    {
        public IEnumerable<News> News { get; set; }
        public IEnumerable<Speaker> Speakers { get; set; }
    }
```

و در اکشن متد Index هم بدین صورت Modelها را به ViewModel ساخته شده پاس می‌دهیم :

```
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 یک View را به صورت Strongly Typed از نوع ViewModel خود که در اینجا SpeakerAndNewsViewModel است ایجاد می‌کنیم :

و View مربوطه را به شکل زیر تغییر می‌دهیم :

```
@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: ۱۳۹۱/۱۲/۱۳ ۱۴:۴۶

· 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 لزوما برای این نیست از چند موجودیت دیتابیسی یک شیء بسیازیم 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: ۱۳۹۱/۱۲/۱۴ ۱۰:۵

· 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.
مثلا 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 Finde(long personId);



}
```

من حرف شما رو بد برداشت کردم یا اینکه دیدم نسبت به مدل اشتباه بوده؟

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۱/۱۲/۱۴ ۱۰:۴۲

ببنید Model به گفته آقای Scott Allen ، در توسعه نرم افزار میتونه صدها مفهوم رو پوشش بده، در کل Model رو میتونیم به این صورت تعریف کنیم : مجموعه ایی از کلاسها که تعریف کننده داده‌ها و همچنین Business Ruleها هستند Business Rule هم همون قواعد یا Business Logic مربوطه هستند که در بالا تعریف شده که مشخص میکنه دیتا چطور میتونه ذخیره بشه و یا تغییر پیدا کنه.

نویسنده: محمد آزاد
تاریخ: ۱۳۹۱/۱۲/۱۴ ۱۱:۳۱

برای جدا سازی کد بهتر نیست که مثلا Business نرم افزار تو یه لایه جدا پیاده سازی بشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۱۴ ۱۲:۱۶

البته عنوان کردند که در این زمینه بین علما اختلاف هست.
برای مثال الگویی وجود دارد به نام Active record که دقیقا با تعریف عنوان شده مطابقت دارد (یک شیء می‌تواند متدهای مثلا افزودن و به روز رسانی و غیره را هم داشته باشد). مزایا و معایب خاص خودش را هم دارد. مطالعه در مورد این‌ها جهت آشنایی با سلایق و دیدگاه‌های مختلف طراحی خوب است. اما وجود آن‌ها به معنای الزام استفاده از روش‌های مطرح شده نیست.

نویسنده: امیر بختیاری
تاریخ: ۱۳۹۱/۱۲/۱۴ ۱۷:۳۲

با تشکر از مطلب خوبتان
فقط یه سوال اینکه View Model 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 View Model 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 View Model این وسط برای ارتباط داشته باشیم می‌خواستم ببینم آیا درسته و دلیلش چیه ؟

نویسنده: شاهین کیاست
تاریخ: ۱۳۹۱/۱۲/۱۴ ۱۷:۴۰

سلام ؛
برای درک بهتر ViewModel و علت استفاده از آن این مطلب را مطالعه بفرمایید .

نویسنده: یزدان
تاریخ: ۱۳۹۲/۰۱/۱۲ ۲:۱۶

لطف میکنید نحوه ارتباط viewmodel رو زمانی که بین model و view 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 برام اهمیت داره که متوجهش نمیشم .
با تشکر

نویسنده: علي
تاریخ: ۱۳۹۲/۰۱/۱۲ ۱۰:۳۴

برای اتصال ViewModel به مدل می‌نویسیم: یک خاصیت viewmodel ،خودت رو به یک خاصیت model انتساب بده پدرجان!
همین! (البته حالت پیشرفته‌تر استفاده از AutoMapper هست که نخوای این‌ها رو اگه سختت 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 عموما داره که در بانک اطلاعاتی نیست و فقط در View تعریف شده.

نویسنده: 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**

نویسنده: علي
تاریخ: ۱۳۹۲/۰۱/۱۲ ۱۲:۲۳

انتساب از طریق عملگر مساوی انجام میشه. مثلا user.Name = userViewModel.Name

نویسنده: یزدان
تاریخ: ۱۳۹۲/۰۱/۱۲ ۱۳:۸

این انتساب کجا انجام میشه ؟ توی اکشن ؟
اگه مقدوره یک مثال ساده با ViewModel و Model و مکانی که این انتساب صورت میگیره بیان کنید .

نویسنده: علي
تاریخ: ۱۳۹۲/۰۱/۱۲ ۱۳:۱۳

در همین مقاله بالا از قسمت «اجازه دهید مثالی را در این رابطه براتون بیارم :» به بعد رو مطالعه کنید. این مثال در مورد HttpGet هست. در حالت HttpPost هم همینه. فرقی نمی‌کنه.

نویسنده: 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 متد Application_Start انجام بگیرد.به طور مثال:

```
protected void Application_Start()
        {
          //مپ کردن مدل به ویومدل
          Mapper.CreateMap<DomainClasses.Product, ProductListVM>();
          //مپ کردن ویومدل به مدل
          Mapper.CreateMap<ProductCreateVM,DomainClasses.Product>();
        }
```

حال فرض بگیر دیتایی داری که میخوای انتصاب بدی به ViewModel به این شکل عمل میکنی:

```
public ActionResult ShowList()
{
var productList = _cotext.Products.ToList();
//انتصاب
 var viewmodel = Mapper.Map<List<domain.Products>, List<ProductListVM>>(productList);
 return View(viewmodel);
}
```

حال برعکس میخوای ViewModel انتصاب بدی به Model :

```
[HttpPost]
        public virtual ActionResult Create(ProductCreateVM product)
```

```
        {
            try
            {
                if (ModelState.IsValid)
                {
                    var model = Mapper.Map<ProductCreateVM, domain.Product>(product);
                    _cotext.Products.Add(model);
                    _cotext.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

و ...

موفق باشی

اگرم خواستی میتونی دستی این انتصاب‌ها را انجام بدی البته من پیشنهاد نمیکنم.

نویسنده: یزدان
تاریخ: ۱۳۹۲/۰۱/۱۲ ۱۴:۵۰

تشکر مهندس پایدار بابت جواب و توضیح کاملی که ارائه دادین

نویسنده: مولانا
تاریخ: ۱۳۹۲/۰۱/۳۰ ۱۶:۴۲

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 یافت 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: ۱۳۹۲/۰۱/۳۰ ۲۰:۴۲

- هر دو مورد رو می‌تونید 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. این‌کار برای بالا بردن سرعت نمایشی سایت انجام شده.

نویسنده: ضیا
تاریخ: ۱۳۹۲/۰۲/۲۲ ۰:۱۸

سلام ...

من تاجاییکه میتونستم چیزهایی که درباره viemodel 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 دارم 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 متد Edit Resume استفاده کردم که متد 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 دوره‌های تخصصی گذرانده"},
                        new {Value = "5", Text = "فرم تخصص‌های 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 == "MartialStatus").ToList();
            ViewBag.MartialStatus = 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 دوره‌های تخصصی گذرانده";
    ViewBag.Part5 = "فرم تخصص‌های 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');
```

```
            }
            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 تاهل")
         @Html.DropDownList("وضعیت تاهل", 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)ViewBag.Ability)
                    @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) ViewBag.Ability)
                    @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)
ViewBag.Ability)
                    @Html.ValidationMessageFor(model => item.Knowledge_Of_English_ID)
                </td>
            </tr>

            <tr>
```

```
                <td>
                    @Html.Label("آشنایی با زبان عربی")
                </td>
                <td>
                    @Html.DropDownListFor(model => item.Knowledge_Of_Arabic_ID, (SelectList)
ViewBag.Ability)
                    @Html.ValidationMessageFor(model => item.Knowledge_Of_Arabic_ID)
                </td>
            </tr>

            <tr>
                <td>
                    @Html.Label("آشنایی با ماکروسافت آفیس")
                </td>
                <td>
                    @Html.DropDownListFor(model =>item.Knowledge_Of_Office_ID, (SelectList)
ViewBag.Ability)
                    @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)
ViewBag.Ability)
                    @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)
ViewBag.Ability)
                    @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)
ViewBag.Ability)
                    @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)
ViewBag.Ability)
                    @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)
ViewBag.Ability)
                    @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)
ViewBag.Ability)
                    @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)
ViewBag.Ability)
                    @Html.ValidationMessageFor(model =>item.Public_Relationship_ID)
                </td>
            </tr>

            <tr>
                <td>
                    @Html.Label("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")
}
```

و مشکل اینجاست که بعد از ثبت اطلاعات وقتی به متد post میره مقدارش null هستش ... درحالیکه فقط در صورت edit اینجوریه وقتی از همین viewmodel برای مشاهده رزومه که فقط گزارشگیریه استفاده میکنم نتیجه را میبینم ولی اینجا نه !
متد پست بصورت زیر هستش ... اگر میدونید چطور و چی کار کنم که این درست شه ممنون میشم ... چون دیگه نمیدونم تو متد پست چی بنویسم ... منتظر جوابم ... که چرا null میده و اینکه تو متد پست چطور اینارو ذخیره کنم توی 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 داده‌های 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 ViewModel است تا به خواص آن به صورت strongly typed در یک View دسترسی داشته باشید و تحت کنترل کامپایلر هم باشد (تمام ViewBagها رو حذف و تبدیل به خواص ViewModel کنید).
- کلاس ViewModel ایی که تدارک دیدید، دارای سازنده‌ای است که باید مقدار دهی شود پیش از استفاده. خوب ... MVC از کجا باید تشخیص بده که در زمان Post اطلاعات به سرور چطور باید این‌رو به صورت خودکار وهله سازی کنه؟ به عبارتی ViewModel شما باید بدون سازنده، قابل وهله 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 Model انتساب بدی . بعد این وهله Add شود به Context و دست آخر هم ذخیره.
(البته کار شما به نظر لایه بندی صحیحی نداره وگرنه 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 کودوم 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 متد 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 11 قسمتی در سایت هست. بحث مفصلی است و نیاز به پایه تئوری داره.
- مشکل بعدی کدهای شما به Html.BeginForm بر می‌گرده. زمانیکه پارامتری رو عنوان نکنید به اکشن متد پیش فرض مشخص شده در سیستم مسیریابی اشاره می‌کنه. بهتره که این موارد رو صریحا ذکر کنید تا مشکل نداشته باشید؛ مانند کد زیر. به علاوه شما نمی‌تونید کل یک حلقه رو داخل یک فرم قرار بدید. 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 ResumeOfViewModel شما که در عمل یک لیست است. بهتر است داخل حلقه این View شلوغ رو Refactor کنید به یک Partial view با مدلی از جنس Resume تا بهتر بتونید مسایل رو مدیریت کنید.

```
@using (Html.BeginForm(actionName: "EditResume", controllerName: "SomeName"))
{
}
```

نویسنده: ضیا
تاریخ: ۱۳۹۲/۰۲/۲۲ ۱۳:۵۸

مرسی از جوابتون ...
اینجا یه حلقه هست با اینکه باید 6تا حلقه باشه چون اطلاعات از 6 جدول جدا میاد برای ویرایش شدن با اینکه اینجا فقط مقادیر یه جدول نشون میده 5تا جدول دیگه مونده ... در اینصورت هم از partialview استفاده کنم ؟
اگر آره ... لینکی که دراینباره بتونه من رو کمک کنه دارید ؟(خوندن اطلاعات و ...)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۲۲ ۱۴:۴

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: ۱۳۹۲/۰۴/۲۳ ۲۰:۲۵

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: ۱۳۹۲/۰۴/۲۴ ۹:۳۸

روش با سربار 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: ۱۳۹۲/۰۴/۲۴ ۱۳:۵۵

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: ۱۳۹۲/۰۴/۲۴ ۱۴:۴

یک رای‌گیری در این مورد در حال انجام است

« در ASP.NET MVC آیا ViewModel همان DTO است؟ »

نویسنده: محمد آزاد
تاریخ: ۱۳۹۲/۰۴/۲۴ ۱۴:۱۵

یعنی جواب قطعی براش نیست؟چون رای گیری نمیتونه ملاک خوبی باشه بنظرم

نویسنده: ح مراداف
تاریخ: ۱۳۹۲/۱۱/۲۴ ۱۳:۴۴

با سلام؛ لطفا درباره section speakers@ یکم توضیح بدین. این Section@ به چه کار آید ؟
حدس من اینه که با این دستور بشه توی یک مدل کلی بخش (مثل ماژول) رو تعریف کرد و توی یک ویو بتونیم در جاهای مختلف این Sectionها رو نمایش بدیم ؟ (عدم نیاز به چند Partial View و نوشتن همه توی یک ویو)

نویسنده: وحید نصیری

تاریخ: ۱۳۹۲/۱۱/۲۴ ۱۳:۵۳

در قسمت « تزریق نواحی ویژه یک View در فایل layout » به این موضوع پرداخته شده.

نویسنده: مجتبی فخاری
تاریخ: ۱۳۹۲/۱۲/۱۵ ۰:۲

با سلام؛ برای استفاده از ViewModel در زمان اجرا بعد از این که روی دکمه جدید کلیک می‌کنم این خطا را می‌دهد:

```
Either ErrorMessageString or ErrorMessageResourceName must be set, but not both.
```

میشه راهنمایی کنید؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۱۵ ۰:۱۱

عین خطا را در گوگل جستجو کنید. در اولین لینک پاسخ را پیدا می‌کنید.

نویسنده: منصور جعفری
تاریخ: ۱۳۹۲/۱۲/۲۶ ۲۰:۵۷

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.Enities.Product', but this
dictionary requires a model item of type 'WebAli.ViewModel.ProductCommentListViewModel'.
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۲۶ ۲۳:۱۲

این خطا به این معنا است که در اکشن متد کنترلر 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 متد را بررسی کنید).

حتی در حالت RenderPartial هم ممکن است این خطا 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: ۱۳۹۳/۰۲/۰۲ ۸:۳۳

سلام، من توی مدل پراپرتی از نوع DateTime? دارم که توی view اون رو با کنترل datepicker نمایش میدم. مشکل اینجاست که مقداری که به مدل پاس میشه، تاریخ شمسی هستش و مدل انتظار datetime رو داره. نهایتا توی uow.SaveChanges با این خطا 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: ۱۳۹۳/۰۲/۰۲ ۹:۳۶

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 میشه اما هنگامی که از ViewModel تعریف شده در لایه Presentation استفاده می‌کنم، Validation با رفرش صفحه انجام میشه. دلیلش چیه؟

نویسنده: سروش
تاریخ: ۱۳۹۳/۰۷/۲۱ ۱۹:۵۸

با سلام و درود به شما عزیزان
من یه مشکلی دارم و اون هم اینکه من می‌خواهم یک select new روی مدلم بزنم ولی چون نام فیلدها جدید به همیم خاطر در Partial View یا View اون ها رو 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 هست که بصورت Strongly Type به 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
مرسی

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 مثل الان anonymous type باشه). نوع View رو هم از نوع لیستی از همین ViewModel جدید تعیین کن. به این نکته LINQ 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="",
                 };
```

عنوان: **مخفی کردن کوئری استرینگ‌ها در ASP.NET MVC توسط امکانات Routing**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۱۵ ۱۱:۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, SEO

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
```

سؤال: آیا می‌شود این لینک‌ها را کمی زیباتر و SEO Friendlyتر کرد؟

برای مثال آن‌را به نحو زیر نمایش داد:

```
http://localhost:1036/Home/Details/0/0
```

پاسخ: بلی. 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}", //تطابق با یک چنین مسیرهایی
                defaults: new
                          {
                                controller = "Home", //کنترلری که این نوع مسیرها را پردازش خواهد کرد
                                action = "Details", // اکشن متدی که نهایتا پارامترها را دریافت می‌کند
                                issueId = UrlParameter.Optional, //این خواص نیاز است هم نام پارامترهای اکشن متد تعریف شوند
                                projectId = UrlParameter.Optional
                          }
            );

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

در اینجا یک route جدید به نام دلخواه IssueDetails پیش از route پیش 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 issueId اکشن متدی به نام Details و کنترلر Home ارسال خواهد کرد؛ به همین ترتیب در مورد projectId 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 شکل زیر را خواهد 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 Urlها را دستی ایجاد کنید، تغییری را مشاهده نخواهید کرد.

نظرات خوانندگان

نویسنده: MehRad
تاریخ: ۱۳۹۱/۱۲/۱۵ ۲۰:۲۹

ممنونم از مطلب خوبتون
در بعضی از پروژه‌ها که به صورت ماژول مانند هستند همانند Orchard1 میبینیم که برای هر قسمت یک قسمت مدیریت در نظر گرفته شده
اما آدرس مدیریت هر کدام در قسمت مدیریت بدین شکل میباشد
مثلا ماژول تگ‌ها : که شامل 2 کنترلر یکی خود ماژول و دیگری مدیریت ماژول
اما قسمت مدیریت http://localhost:30320/OrchardLocal/Admin/Tags بدین شکل است
سوال اینجاست در اینجا بر اساس نام ماژول آن را به قسمت مدیریت آن هدایت میکند یا اینکه با امکانات مسیریابی؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۱۵ ۲۱:۱

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 متد موجود در آن. برای توضیحات بیشتر مراجعه کنید به قسمت چهارم سری MVC .

نویسنده: محمد آزاد
تاریخ: ۱۳۹۲/۰۴/۲۲ ۱۱:۶

آیا این امکان هست که یکی از پارامتر هارو نمایش ندیم مثلا به این صورت بشه

```
url: "Details/{projectId}"
```

```
@Html.ActionLink("detail","Details",new {issueId=10,projectId=11})
```

من این تغییر رو ایجاد کردم و می‌خوام مقدار issueId رو نمایش ندم اصلا اما url ساخته شده این شکلی میشه

```
http://localhost:2435/Home/Details?issueId=10&projectId=11
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۲۲ ۱۱:۳۱

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 متد Details هم باید بر همین اساس و ترتیب تغییر کند.
سپس اگر نمی‌خواهید یکی از پارامترها را ذکر کنید؛ پارامترهای این متد را nullable تعریف کنید. به این ترتیب با مقدار دهی null به issueId در ActionLink، خودبخود در لینک تولیدی نهایی ظاهر نخواهد شد.

نویسنده: محمد آزاد
تاریخ: ۱۳۹۲/۰۴/۲۲ ۱۴:۲۴

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 http://mysite/Project/Detail /ProjectId ولی می‌خوام به این صورت نمایش داده بشه

http://mysite/Project/Detail /ProjectTitle آیا راهی برای این وجود داره ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۲۲ ۱۴:۴۸

برای تکمیل بحث به مطلب « سازگار کردن لینک‌های قدیمی یک سایت با ساختار جدید آن در ASP.NET MVC » مراجعه کنید؛ همچنین مطالعه پیشنیاز آن هم مفید است.

نویسنده: رضا منصوری
تاریخ: ۱۳۹۲/۰۷/۰۶ ۱۰:۱۶

ممنون از مطلب عالیتون یه سوال کوچیک
شما تو سایتتون از نوشته‌های انگلیسی برای روتینگ صفحاتتون استفاده کردید مثل http://www.dotnettips.info/newsarchive

به نظر شما اگه فارسی باشه از نطر سئو بهتره یا نه مثل این
http://www.dotnettips.info /مقالات-جدید

یا با این کار مشکلات دیگه ای بوجود میاد ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۷/۰۶ ۱۰:۲۷

اکثر خزنده‌ها با آدرس‌های فارسی مشکل دارند. اگر لاگ‌های خطای سرور رو بررسی کنید عموما عاجزند از درک encoding این نوع urlها و مدام به خطای 404 برخورد می‌کنند. برای نمونه این تصویر خروجی یکی از همین خزنده‌هایی است که سایت را برای بودن آدرس یا جستجوی خاصی در سایت مدام چک می‌کند. به نظر Bing است.

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 خزنده در تفسیر آن اشتباه کرد، به خطای 404 برخورد نکند.

نویسنده: رضا منصوری
تاریخ: ۱۳۹۲/۰۷/۰۶ ۱۲:۳

بازم یه سوال دیگه
این صفحه‌ی اصلی یک سایته http://www.beatskala.ir/

وقتی روتینگ انجام میدیم انگار فایل‌های جی کوئری لود نمیشه یعنی اینجوری میشه(منوها کار نمیکنه)
http://www.beatskala.ir/Post/7/%DA%86%DA%AF%D9%88%D9%86%D9%87-5

به نظر شما مشکل کجاست ممنون

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: (^)

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}", //تطابق با یک چنین مسیرهایی
    defaults: new
    {
        controller = "Post", //کنترلری که این نوع مسیرها را پردازش خواهد کرد
        action = "Manage", // اکشن متدی که نهایتا پارامترها را دریافت می‌کند
        page = UrlParameter.Optional //این خواص نیاز است هم نام پارامترهای اکشن متد تعریف شوند
    }
);
```

Or in AdminAreaRegistration

```
context.MapRouteLowercase(
                name: "myroute",
                url: "Manage/{page}", //تطابق با یک چنین مسیرهایی
                defaults: new
                {
                    controller = "Post", //کنترلری که این نوع مسیرها را پردازش خواهد کرد
                    action = "Manage", // اکشن متدی که نهایتا پارامترها را دریافت می‌کند
                    page = UrlParameter.Optional //این خواص نیاز است هم نام پارامترهای اکشن متد تعریف
شوند
                }
            );
```

نویسنده: محمدرضا برنتی
تاریخ: ۱۳۹۲/۰۹/۳۰ ۲:۲۸

حل شد :

```
context.MapRouteLowercase(
                "Admin_default2",
                "Admin/Post/Manage/{page}",
                new { controller = "Post", action = "Manage", page = UrlParameter.Optional }
            );
```

ممنون ...

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؟ اصلا پیشنهاد میشه یا روش مناسبی نیست، تاثیر زیادی توی سرعت میزاره؟برای اینکار پیشنهادتون چیه ؟
ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۱۶ ۱۳:۸

« هدایت درخواست فایل‌های استاتیک در ASP.NET MVC به یک کنترلر »

عنوان: **توزیع پروژه‌های ASP.NET MVC بدون ارائه فایل‌های View آن**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۱۶ ۲۲:۳۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

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 آن‌را 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 نمائید:

```
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 خواهند 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.

Solution 'MvcViewGenTest2' (1 project)
MvcViewGenTest2
Properties
References
App_Data
App_Start
FilterConfig.cs
RazorGeneratorMvcStart.cs
RouteConfig.cs
WebApiConfig.cs
Controllers
HomeController.cs
Models
Views
Home
Index.cshtml
Index.generated.cs
Web.config
Global.asax
packages.config
Web.config

Properties
Index.cshtml File Properties
Advanced
Build Action
Content
Copy to Output Directory
Do not copy
Custom Tool
RazorGenerator
Custom Tool Namespace

بدیهی است اگر از دستور Enable-RazorGenerator استفاده نکنید، نیاز خواهید داشت تا تنظیم گزینه Custom tool به RazorGenerator 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(ی 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 موکول 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 Wiki پروژه Razor Generator مطالعه کنید. انتقال Viewها به یک اسمبلی دیگر هرچند در این روش کاملا ممکن شده و کار می‌کند اما صفحه dialog افزودن یک view جدید مهیا در کلیک راست بر روی اکشن متدهای 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 مجددا به روز و تولید خواهند شد.

فایل‌های Helper قرار گرفته در پوشه App_Code

اگر HTML Helperهای خود را توسط فایل‌های Razor قرار گرفته در پوشه App_Code تولید می‌کنید، پس از اجرای دستور Enable-RazorGenerator، برای این موارد نیز فایل‌های cs متناظری تولید می‌شود. با این تفاوت که Build Action آن‌ها بر روی Compile 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 مثلا:

```
@using MvcViewGenTest2.app_code
```

البته با استفاده از Razor Generaor دیگر نیازی به استفاده از پوشه App_Code نخواهد بود؛ از این جهت که کار کامپایل خودکار، به زمان اجرا موکول نخواهد شد. بنابراین اینبار در هر جایی که علاقمند بودید می‌توانید این فایل‌های 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 تغییر زیر را اعمال نمائید:

```
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:

```
<img title="ارسال ایمیل به دوستان" alt="email" src="~/content/images/gmail.png" />
```

مسیر فایل‌ها حتما باید توسط Url.Content مشخص شود:

```
<img title="ارسال ایمیل به دوستان" alt="email" src="@Url.Content("~/content/images/gmail.png")" />
```

نویسنده: Green
تاریخ: ۱۳۹۱/۱۲/۲۳ ۱۸:۱۶

اتفاقا چند وقت پیش با RazorGenerator کار کردم اونموقع مطلب شما رو در این باره نخونده بودم با کلی مکافات تونستم ازش خروجی بگیرم که یه سوال برام پیش اومد 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 میشه) هربار «از ابتدا» 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 واقعا 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.

نویسنده: مولانا
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؟

نویسنده: وحید نصیری
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.

نویسنده: سمیرا قادری
تاریخ: ۱۳۹۲/۰۴/۰۲ ۱۶:۸

با سلام
چرا بعد از publish دوباره view(ها chtml) ارسال می‌شوند . در صورتی که تمامی کارهای گفته شده در بالا انجام شده ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۰۲ ۱۷:۴۵

publish برای روال معمول و استاندارد کار است. کاری که در اینجا انجام شده غیرمعمول است. مدیریت نهایی انتشار آن دستی خواهد بود (در کل پوشه‌های اسکریپت، bin، content و چند فایل config باید توزیع شوند + نکته تکمیلی 2 که در بالا ذکر شده).

نویسنده: محسن
تاریخ: ۱۳۹۲/۰۷/۱۶ ۱۷:۱۰

سلام آقای نصیری.من نسخه Razor Generator 2.1.2 رو نصب کردم.اما دستور enable-razorgenerator بدون دادن هیچ خطایی انجام می‌شود اما فایلهای Cs مربوط به Viewها تولید نمیشه و وقتی روشون Run Custom Tool رو می‌زنم این پیغام میاد.ممنون اگه راهنماییم کنید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۷/۱۶ ۱۷:۱۸

قسمت (ب) ابتدای بحث کافی نیست. قسمت (الف) توضیح داده شده نیز باید انجام شود .

نویسنده: محمد آزاد
تاریخ: ۱۳۹۴/۰۲/۰۲ ۰:۱۶

آیا بین این روش و portable area تفاوتی هست؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۴/۰۲/۰۲ ۰:۴۴

- razor generator توسط اعضای تیم نیوگت و همچنین ASP.NET MVC تهیه شده.
- razor generator برخلاف portable areas در زمان کامپایل کار parse و همچنین قرار دادن viewها را در DLL مرتبط انجام می‌دهد.
- آخرین به روز رسانی portable areas در سال 2010 بوده، اما razor generator مدام به روز می‌شود.

عنوان: **حذف هدرهای مربوط به وب سرور از طریق برنامه نویسی**
نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۱/۱۲/۱۹ ۲۳:۲۰
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, IIS, ASP.Net MVC, MVC, Security

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.

برای یک وب اپلیکیشن APS.NET MVC این هدرها را داریم :

Server: که توسط IIS اضافه میشود.

X-AspNet-Version: که در زمانFlush در httpresponse اضافه میشود.

X-AspNetMvc-Version: که توسط MvcHandler در System.Web.dll 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 Global.asax.cs به رویداد Application_Start 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 المنت 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 ورژن فریم ورک نمی‌باشد.

نظرات خوانندگان

نویسنده: sysman
تاریخ: ۱۳۹۲/۰۱/۱۴ ۰:۲۲

ممنون از این مطلب مفید ولی من یک مشکلی دارم کاری که گفتید را انجام دادم وهمه چیز خوب انجام شد ولی در بخش GET modernizr-2.5.3.js باز هم اطلاعات سرور رو به من میدهد

نویسنده: علي
تاریخ: ۱۳۹۲/۰۱/۱۴ ۰:۳۶

فایل‌های استاتیک رو هم باید به موتور ASP.NET مپ کنید. تا زمانیکه مپ نباشند مستقیما توسط IIS سرو می‌شن و تنظیمات ASP.NET روی اون‌ها تاثیری نداره.

نویسنده: sysman
تاریخ: ۱۳۹۲/۰۱/۱۴ ۱۰:۴۰

نمی دونم این کاری که گفتین رو دقیقا چطور انجام بدهم 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 ^)
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 مستقیما و بدون دخالت ASP.NET، توسط IIS توزیع نمی‌شوند.
- برای مپ 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 ازش استفاده کردم ولی موقع حذف Header مربوط به سرور ارور زیر رو میده:
This operation requires IIS integrated pipeline mode.
ضمنا من از iis 7.5 استفاده می‌کنم و توی Application Pools هم .Net AppPool Classic رو در حالت pipeline mode:Integrated قرار دادم هم DefaultAppPool ولی همچنان ارور رو دارم.مشکل از کجاست؟
ممنون.

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، اصلا نیازی به کدنویسی برای حذف هدرهای وب سرور نخواهید داشت. به قسمت HTTP response headers کنسول مدیریتی مراجعه و مداخل موجود را حذف یا ویرایش کنید.
+ 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 C:\Windows\System32\inetsrv\urlscan\UrlScan.ini مراجعه و RemoveServerHeader را با 1 مقدار دهی کنید. ضمنا 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

با 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>
```

نویسنده: 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 کانفیگ میتونید تغییرش بدید:

```
<system.web>
    <sessionState cookieName="foo" />
</system.web>
```

عنوان: **ایجاد helper برای Nivo Slider در Asp.net Mvc**
نویسنده: علی حق جو
تاریخ: ۱۳۹۲/۰۱/۲۹ ۱۶:۳۵
آدرس: www.dotnettips.info
برچسب‌ها: ASP.Net MVC, jQuery, NivoSlider, Mvc helper, MVC

کامپوننت‌های jQuery 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
```

```
StyleBundle("~/Content/NivoSliderDefaultTheme").Include("~/Content/nivoSlider/themes/default/default.cs
s"));
   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(",pauseOnHover:{0}", _pauseOnHover.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("$('#" + _id + "').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; }
    }
```

-6سپس 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 = "",
```

```
                    },
                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 NivoSliderهای 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 (deafult- 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. دریافت 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 پیداش کنم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۰۸ ۱۱:۴۷

این dll جزو فایل‌های استاندارد یک پروژه ASP.NET MVC **4** است. در اینجا پیوست نشده، چون نیازی نیست هربار 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.

عنوان: آشنایی با Fluent Html Helpers در MVC
نویسنده: حمیدرضا صادقیان
تاریخ: ۱۳۹۲/۰۱/۳۰ ۱۸:۲۰
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net MVC, Fluent Html Helpers, MVC

همان طور که قبلا نیز اشاره شد اینجا به صورت خلاصه هدف Fluent 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 helperها شرح دهیم.
در اینجا مثالی رو شرح میدهیم که در آن کنترل هایی از جنس input به صورت helper ساخته و برای فرستادن ویژگی‌های اچ تی ام ال (HTML Attributes) آن از Fluent Html Helpers بهره میگیریم بدیهی است که از این Fluent میتوان برای helperهای دیگر هم استفاده کرد.
در ابتدا یک نگاه کلی به کد ایجاد شده می‌اندازیم :

```
@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 می‌شود ، همان طور که مشاهده می‌کنیم تمام ویژگی‌ها 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 می‌رویم ، این کلاس باید از کلاس RouteValueDictionary ارث بری داشته باشد، این کلاس یک دیکشنری برای ما آماده می‌کند تا مقادیرمون رو در آن بریزیم در اینجا برای ویژگی‌ها و مقادیرشون.

```
public class Attributes : RouteValueDictionary    {
        /// <summary>
        /// Configures this instance.
        /// </summary>
        /// <returns></returns>
        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;
        }
    }
```

متد استاتیک Configure همیشه به عنوان شروع کننده fluent و از اون برای ساختن یک وهله 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 ویژگی‌های 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

عنوان: **نحوه استفاده از افزونه Firebug برای دیباگ برنامه‌های ASP.NET مبتنی بر jQuery**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۰۴ ۱۲:۳۰
آدرس: www.dotnettips.info
برچسب‌ها: ASP.Net, Debugging, MVC, FireBug, jQuery, jQuery Ajax

هر از چندگاهی سؤال «این مثال jQuery رو 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 Image Slider را که پیشتر در سایت مطرح شده است، به نحوی دیگر بررسی خواهیم کرد:

1) فراموش می‌کنیم تا اسکریپت اصلی jQuery را به درستی پیوست و مسیردهی کنیم.
2) مسیر Generic handler دیگری را ذکر می‌کنیم.
3) مسیرهای تصاویری را که Image slider باید نمایش دهد، کاملا بی‌ربط ذکر می‌کنیم.
4) خروجی JSON نامربوطی را بازگشت می‌دهیم.
5) یکبار هم یک استثنای عمدی 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 Content، Scripts 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، 5 مورد مشکل ساز یاد شده لحاظ شده‌اند)
محتویات فایل OrbitHandler.ashx.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 عمدا از یک لیست جنریک تشکیل شده در حافظه کمک گرفته شده است. تصاویر برنامه در پوشه Images واقع در ریشه سایت، قرار دارند. بنابراین توسط متد PicturesDataSource، فایل‌های این پوشه را یافته و مطابق ساختار کلاس Picture بازگشت می‌دهیم. نهایتا این اطلاعات به ظاهر قرار است با فرمت JSON بازگشت داده شوند تا بتوان نتیجه را توسط افزونه Orbit استفاده کرد.

همچنین کدهای صفحه ASPX ایی 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 = '<img src="' + b.Path + '" alt="' + b.Title + '"/>';
                        $("#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 مجددا 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 خطاها، بر روی برگه Console آن کلیک کنید:

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:

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.

اگر به برگه کنسول دقت 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.ashx.cs مراجعه کرده و سطر throw new InvalidDataException را حذف می‌کنیم. در ادامه برنامه را کامپایل و مجددا 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 پوشه 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 jQuery Ajax 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 اسکریپتی 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 فرمت 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:

Console HTML CSS Script DOM Net Cookies YSlow Pixel P... Firefi

Edit | body < html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1
<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;">
          <img alt="01.png" src="01.png" style="z-index: 1; display: block; left:
          -48px;"> jQuery15100924838756054342 2=Object { olddisplay="inline" }
          <img alt="02.jpg" src="02.jpg" style="display: block; left: -48px;
          z-index: 1;">
          <img alt="03.jpg" src="03.jpg" style="display: block; left: 0px; z-index:
          3;">
        </div>
```

نظرات خوانندگان

نویسنده: صادق
تاریخ: ۱۳۹۲/۰۲/۰۴ ۱۲:۵۱

ممنون آقای نصیری - برای کروم چطور، آیا developer tools کفایت میکند یا ابزار بهتری سراغ دارید؟
البته من گاهی که نیاز به دستکاری ریسپانس و ریکوئست باشه از فیدلر استفاده میکنم

نویسنده: 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 تقریبا یکی است.

نویسنده: 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.

نویسنده: Abolfazl
تاریخ: ۱۳۹۲/۰۲/۰۸ ۲۳:۴۸

سلام آقای نصیری ..ممنونم ..خیلی عالی بود
من قبلا هم این مشکل را داشتم و نمی‌تونستم برطرفش کنم ..الان درست شد..مرسی

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۲/۰۹ ۱۶:۴۲

```
$(document).ready(function () {
    $(':radio').click(function () {
        debugger; //فعال سازی دیباگر ویژوال استودیو
     });
});
```

یک نکته جانبی است برای فعال سازی دیباگر خود ویژوال استودیو در حین کار با جی‌کوئری

نویسنده: ahmad.valipour
تاریخ: ۱۳۹۲/۰۲/۱۴ ۱۲:۱۹

با 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؟

نویسنده: 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 سطری break point قرار می‌دید و خروجی

یک کوئری را بررسی می‌کنید، این دیباگر 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 خطایی وجود ندارد ....
ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۲۹ ۱۱:۲۹

حداقل دو علت می‌تونه داشته باشه:
الف) تصاویر رو نمی‌تونه پیدا کنه، یا صفحه کش شده بیش از حد. قسمت «اجرای کدهای jQuery Ajax 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 /* گاهی از اوقات خصوصا برای آی ای 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 متد 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 همانطور که قبلا 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 میتونید کمکم 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: ۱۳۹۲/۰۷/۰۷ ۱۱:۲۸

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 ایی با این مشخصات.

عنوان: عدم امکان تغییر اطلاعات مدل در HTML Helpers پس از Postback در ASP.NET MVC
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۱۰ ۱۲:۳۰
آدرس: www.dotnettips.info
برچسب‌ها: MVC

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 متد ModelState.Clear
این متد کلیه داده‌های موجود در ModelState را منجمله خطاهای حاصل از اعتبارسنجی، 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 متد 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 متد 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" />
```

عنوان: **اضافه کردن Watermark به تصاویر یک برنامه ASP.NET MVC در صورت لینک شدن در سایتی دیگر**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۱۹ ۸:۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, Security, Watermark

درگیر شدن با سایت‌های 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 نشدن آن‌را باید زد. اما ... می‌توان همین سایت‌ها را تبدیل به تبلیغ کننده‌های رایگان کار خود نمود که در ادامه نحوه انجام آن‌را در یک برنامه ASP.NET MVC بررسی خواهیم کرد:

الف) نیاز است ارائه تصاویر تحت کنترل برنامه باشند.

```
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); // تمیز سازی امنیتی 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 متد Image 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، واقعا یک fileName صحیح باشد، نیاز است توسط متد 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 متد نیز به نحو زیر است:

```
<img src="@Url.Action(actionName: "Image", controllerName: "Home", routeValues: new { fileName =
"EF_Stra_08.gif" })" />
```

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 مرورگر صورت می‌گیرد. اگر Host این UrlReferrer با Host درخواست جاری یکی بود، یعنی تصویر از سایت خودمان فراخوانی شده‌است.

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 WebImage که کار افزودن Watermark را بسیار ساده کرده است. نمونه‌ای از نحوه استفاده از آن‌را در متد فوق ملاحظه می‌کنید.
اما ... پس از امتحان تصاویر مختلف ممکن است گاها 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 فرمت 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 متد Image 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

اتفاقا دیروز دیدم در سایتی مطالب این سایت استفاده شده بود که برایم خوشایند نبود. توجهم به تصاویر جلب شد که نام dotnettips درج شده بود فکر کردم که جدیدا این امکان اضافه شده و بعد از بررسی متوجه شدم که این اتفاق در سایتهای 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 اصلا 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 وب فرم‌ها (فایل‌های ashx) هستند. در اینجا بجای Url.Action فوق برای مقدار دهی img src، خواهید داشت pic.ashx?filename=tst.png.
ضمنا ASP.NET MVC سورس باز است و کدهای متد AddTextWatermark آن‌را در اینجا می‌توانید مطالعه کنید.

عنوان: **چگونگی رسیدگی به Null property در AutoMapper**
نویسنده: مهراد
تاریخ: ۱۳۹۲/۰۲/۲۲ ۱۷:۴۰
آدرس: www.dotnettips.info
گروه‌ها: C#, MVC, Mapper, AutoMapper

AutoMapper کتابخانه‌ای برای نگاشت اطلاعات یک شیء به شیءایی دیگر به صورت خودکار می‌باشد.

در این مقاله چگونگی رسیدگی به Null property را 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 مبدا، 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 است ولی دارای متد اختیاری NullSubstitute 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: ۱۳۹۲/۰۲/۲۳ ۲۰:۴۱

با سلام

مهندس من یه دیتابیس دارم که حاوی اطلاعات است در جداول اون در تمام ستون‌ها به غیر از ستون کلید اومده تیک alow null رو فعال کرده یعنی این ستون‌ها می‌تونه مقدار null رو بگیره .

حالا اون برنامه که این اطلاعات رو وارد دیتابیس کرده اومده هر ستونی که نوعش رشته 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 دیتابیس ذخیره کنم ؟

نویسنده: MehRad
تاریخ: ۱۳۹۲/۰۲/۲۳ ۲۱:۴۶

سلام
بستگی به کار خودتون داره . مطلب بالا مربوط به نگاشت اطلاعات یک شیء به شیءایی دیگره .اما شما برای در نظر گرفتن مقدار پیش فرض در دیتابیس همون طور که میدونید با تنظیم Default Value or Binding میتونی مقدار پیش فرضی برای ستون‌های Null در نظر بگیری.

نویسنده: ناصر پورعلی
تاریخ: ۱۳۹۳/۰۳/۲۱ ۱۶:۴۴

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.
منتهی 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

نویسنده: علیرضا پایدار
تاریخ: ۱۳۹۳/۰۳/۲۲ ۹:۱۵

```
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 باشه مشکلی پیش نمیاد اگر هم null باشه بازم مشکلی پیش نمیاد و VMProperty برابر با null هست.

عنوان: **return File در ASP.NET MVC و نام‌های یونیکد**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۲۸ ۸:۵
آدرس: www.dotnettips.info
برچسب‌ها: MVC

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 نخواهید 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 متد 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 عملا 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.

عنوان: **تولید SiteMap استاندارد و ایجاد یک ActionResult اختصاصی برای Return کردن SiteMap تولید شده**
نویسنده: سید مهران موسوی
تاریخ: ۱۳۹۲/۰۲/۲۹ ۲۱:۱۵
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net MVC, SEO, sitemap, MVC

یکی از itemهای مهم در بهینه سازی SEO یک وب‌سایت وجود یک SiteMap استاندارد متشکل از لینک‌های موجود در سایت هست که در وب‌سایت‌های داینامیک معمولا این لینک‌ها بر اساس داده‌های موجود در بانک اطلاعاتی ایجاد میشه. برای مثال مطالب، اخبار و ....

در اینجا بنده قبلا یک کلاس برای تولید SiteMap آماده کردم که در پروژه‌های خودم ازش استفاده میکنم. توسط این کلاس میتونید به صورت داینامیک SiteMap وب‌سایت مبتنی بر ASP.NET MVC خودتون رو ایجاد کنید.

برای آشنایی با ساختار یک SiteMap استاندارد میتونید به لینک رسمی روبرو مراجعه کنید :
http://www.sitemaps.org/de/protocol.html

بنده کلاس‌های زیر رو بر مبنای لینک مذکور در سایت رسمی SiteMaps تولید کردم. بعد از تولید SiteMap نیاز دارید که اون رو مثلا به عنوان خروجی یک ActionResult بازگردونید. برای این کار هم یک کلاس با نام XmlResult مشتق شده از ActionResult آماده سازی کردم که کلاس تولید شده SiteMap رو Serialize میکنه و به عنوان نتیجه‌ی یک Action باز می‌گردونه .

```
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.Output, 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 هایی که به sitemap اضافه میشه واکشی 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 میتونید داده‌ها رو بر اساس ساختار بانک اطلاعاتی خودتون واکشی 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: ۱۳۹۲/۰۳/۰۱ ۱۴:۳۹

یک نکته تکمیلی:

ایجاد خودکار آدرس sitemap.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: ۱۳۹۲/۰۳/۰۱ ۱۴:۵۶

این کار که 100% باید انجام بشه وگرنه بی فایدست تمام ماجرا . ممنون از آقای نصیری بابت پیوست این نکته. دوستان حتما توجه داشته باشن ....

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۳/۰۱ ۱۶:۴۱

```
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 سایت‌مپ اشاره کنه تا موتورهای جستجو بدونند کجا رو باید بررسی کنند

نویسنده: امیرحسین مرجانی
تاریخ: ۱۳۹۲/۰۵/۲۸ ۲۲:۱۹

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 نمی‌کنه؟

2 -همان طور که می‌دونید حداکثر می‌شود 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 می‌شه رفتار کرد؟

3- و سوال آخر اینکه چرا آقای نصیری لینک‌های محدودی را در sitemap سایت قرار داده اند؟ دلیل خاصی داشته است؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۵/۲۸ ۲۲:۲۸

1 - از OutputCache حتما باید استفاده کنید.

2 و 3 - موتورهای جستجو به دنبال لینک‌های 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 صحت اطلاعات بانک اطلاعاتی موتورهای جستجو).

این مورد را اگر خواستید بررسی کنید، از ابزار وب مستر گوگل بهره بگیرید. هر روز آمار می‌دهد، چه مواردی کلا ایندکس شده

(همچنین چه تعدادی دقیقا 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.

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 را در سایت مپ قرار بدهم و بعد از ایندکس شدن توسط گوگل آن‌ها را تغییر دهم که عملا برای این حجم بالا ممکن نیست.
می تونم بپرسم جرا با ساختن فایل‌های فیزیکی برای سایت مپ موافق نیستید؟
(البته در شرایط صورت مسئله ای که بیان کردم) در این صورت می‌شود طوری برنامه ریزی کرد که این فایل ها(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 هرباره آن‌ها به وجود نیاید.

باز 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 طبیعتا درسته) ولی توی پروژه بنده IIS ایرور می‌ده که این فایل (sitemap.xml)رو پیدا نکرده! فکر می‌کنم عمل routing به درستی انجام نمی‌شه.
من از mvc4 در vs2102 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

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 » احتمالا 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 تعریف شود. route پیش فرض اصطلاحا catch all است و نوبت به بقیه نخواهد رسید.

نویسنده: آیمو
تاریخ: ۱۳۹۲/۰۸/۰۴ ۱۷:۵۹

ایییییییییییییول...! خیلی باحالین. دقیقا 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 مستقیما 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

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۲۲ ۱۵:۳۹

موتورهای جستجو بر اساس تنظیمات فایل robots.txt قرار است آن‌را یافته و صدا بزنند. کاربرد شخصی ندارد.

نویسنده: حمید حسین وند
تاریخ: ۱۳۹۳/۰۶/۳۱ ۲۲:۲۶

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

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۶/۳۱ ۲۳:۰

اکثر کدهای آن یکی است منهای کلاس XmlResult آن.
برای پیاده سازی و جایگزینی آن در وب فرم‌ها یک Generic handler استاندارد را اضافه کنید و در متد ProcessRequest آن، کدهای متد ExecuteResult فوق را قرار دهید.

```
public void ProcessRequest (HttpContext context)
```

البته چون در اینجا HttpContext را مستقیما دارید، قسمت‌های context.HttpContext متد ExecuteResult می‌شوند context.
Sitemap_ashx.cs

نویسنده: حسین
تاریخ: ۱۳۹۳/۰۸/۱۹ ۱۲:۱۰

با تشکر
آیا راهی وجود داره بدون تعریف کردن route بتوان به اکشن مورد نظر دسترسی پیدا کرد؟
چون برای فعال کردن این قابلیت باید اجازه وارد کردن dot در url رو داد که برامون امکان پذیر نیست

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۸/۱۹ ۱۲:۱۵

کمی بالاتر در نظرات عنوان شده، موتورهای جستجو این مسیر را (هر آدرسی با هر شکل یا نامی که دارد؛ مهم نیست) صرفا از فایل robots.txt دریافت می‌کنند.

نویسنده: سانی
تاریخ: ۱۳۹۳/۱۱/۱۳ ۱:۷

سلام؛ عالی بود. من برای خواندن از بانک اطلاعاتی این کد رو نوشتم

```
public virtual ActionResult Sitemap()
        {
            var data = new sunn.Models.ApplicationDbContext().Posts.ToList();
            SiteMap sm = new SiteMap();
            foreach (var siteno in data)
            {
                sm.Add(new Location()
                   {
                       Url = string.Format("http://www.MySite.ir/Develop/Home/Post/{0}", siteno.Id),
                       LastModified = siteno.InsertDate,
                       Priority = 0.5D
                   });
            }
```

```
        return new XmlResult(sm);
    }
```

زمانیکه در آدرس بار نام این اکشن رو میزنم، تمامی اطلاعات رو نشون میده؛ اما بدون هیچ فرمتی و پشت سر هم. اما ViewSource رو که میزنم توی source کاملا نقشه سایت رو نشون میده .

دو تا سوال : اول اینکه به چه نحوی میتونم آدرس فایل xml را بدست بیارم که مثلاً به گوگل معرفی کنم و دوم اینکه به جای نمایش این صفحه ناخوانا بتوانم یک صفحه خواناتر با فرمت نمایش بدم

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۱۱/۱۳ ۱:۱۸

خوانایی این خروجی اهمیتی نداره چون برای ماشین درست میشه نه برای انسان. فرمت هم نداشته باشه بهتره چون حجم کمتری رو به خودش اختصاص می‌ده. ولی در کل باید خاصیت Indent تنظیم بشه، در صورت نیاز.

در مورد خروجی با پسوند xml هم باید مسیریابی جدید اضافه کنی که قبلا در نظرات همین بحث مطرح شده (اولین نظر به علاوه مورد robots.txt که در نظرات بعدی هست).

عنوان: **نحوه ایجاد یک تصویر امنیتی (Captcha) با حروف فارسی در ASP.Net MVC**
نویسنده: پژمان پارسائی
تاریخ: ۱۳۹۲/۰۳/۰۳ ۱۷:۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, ASP.Net MVC, CAPTCHA, Security

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 قبلا آقای پایروند 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 می‌باشد. (اطلاعات بیشتر را اینجا بخوانید) اما ممکن است مواقعی پیش بیاید که بخواهید نوعی را توسط یک اکشن متد برگشت دهید که به صورت توکار تعریف نشده باشد. مثلا 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 ActionResult 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 متد ExecuteResult را override کنید. متد ExecuteResult به صورت خودکار هنگامی که از CaptchaImageResult به عنوان یک نوع برگشتی اکشن متد استفاده شود اجرا می‌شود. به همین خاطر باید تصویر امنیتی توسط این متد 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.PageUnit = 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();
```

متد 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، مثلا اگر عدد 4524 توسط متد CreateSalt تولید شده باشد، رشته "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، متد الحاقی NumberToText با پارامتر Language.Persian وظیفه تبدیل عدد salt را به حروف فارسی معادل خواهد داشت.

به صورت پیش فرض نوشته‌های تصویر امنیتی به صورت چپ چین نوشته خواهند شد، و با توجه به این که نوشته ای 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 نوشته 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

کسینوسی برای نوشتن نوشته‌ها به صورت 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 نظرر، 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 بنقش 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 ware 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 متدی 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 متد مقداردهی شود، آنگاه تصویر امنیتی تولید شده نمایش پیدا می‌کند:

```
<img src="@Url.Action("CaptchaImage")"/>
```

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 مورد 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 امنتی 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

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۰۳ ۱۹:۳۶

ممنون. میشه قسمت بررسی نهایی در اکشن متد رو هم کپسوله کرد (چیزی شبیه به امکانات AOP سرخود در MVC). مثلا یک ویژگی جدید به نام ValidateCaptcha درست کرد که به اکشن متد اعمال شود و کار بررسی صحت اطلاعات ورودی مخصوص Captcha رو انجام و نهایتا اطلاعات ModelState رو بر 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", "لطفا 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
اشتباه وارد کرده اید");
        }
    }
```

به این صورت (با استفاده از ویژگی فوق) همان بررسی متداول ModelState.IsValid در یک اکشن متد کافی خواهد بود.

نویسنده: پژمان پارسائی
تاریخ: ۱۳۹۲/۰۳/۰۴ ۰:۲۷

بسیار عالی و سازنده، خیلی نکته مفیدی بود

نویسنده: میثم خوشقدم
تاریخ: ۱۳۹۲/۰۳/۰۴ ۱۳:۵۷

ممنون - مطلب کاربردی و مفیدی بود

شخصا 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

بازهم تشکر می‌کنم

نویسنده: میثم خوشقدم
تاریخ: ۱۳۹۲/۰۳/۰۴ ۱۵:۱۷

سلام

اگر کد شما را گسترش دهیم و به جای پیغام‌های خطای ارسال شده، مقادیر Attributeها یا متادیتاهایی مثل Required را خواند و مستقیم پیغامی به کنترل و ModelState پاس نداد، فکر می‌کنم کامل‌تر باشه .

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۰۴ ۱۶:۳۸

حالت کاملتر زمانی است که در سازنده کلاس ویژگی، این خطاها رو دریافت کنید و به کاربر اجازه بدید، پیغام‌های دلخواهی رو تنظیم کنه. چون Required فقط یک حالت است. حالت بعدی عدم تساوی مقدار ورودی با مقدار اصلی Captcha است که نیاز به پیغام دومی دارد.

نویسنده: پژمان پارسائی
تاریخ: ۱۳۹۲/۰۳/۰۵ ۸:۴۱

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،
معتبر نبودن session نگهدارنده عدد تصویر امنیتی بعد از یک میزان مشخص،
و غیره

نویسنده: حسینی
تاریخ: ۱۳۹۲/۰۳/۰۶ ۱۹:۴

سلام و مرسی 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.

نویسنده: پژمان پارسائی
تاریخ: ۱۳۹۲/۰۳/۰۶ ۲۱:۲

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.

نویسنده: پژمان پارسائی
تاریخ: ۱۳۹۲/۰۳/۱۵ ۶:۲۸

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

نویسنده: پژمان پارسائی
تاریخ: ۱۳۹۲/۰۳/۲۱ ۲۲:۵۲

پوشه‌های packages، bin، و obj 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 خطایی 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.

نویسنده: بهمن خلفی
تاریخ: ۱۳۹۲/۰۴/۱۱ ۱۰:۳۴

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;
            TimeIsExpired = string.Format("حداکثر مهلت وارد کردن کد امنیتی {0} ثانیه
است",ExpireTimeCaptchaCodeBySeconds.ToString());

        }
```

باز ممنون!

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۲/۰۴/۱۴ ۲۱:۳۸

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 لاگ 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.

نویسنده: 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 رو به وب کانفیگ اضافه کنید تا خطا رو بتونید لوکال دیباگ کنید.

نویسنده: 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
```

نویسنده: پژمان پارسائی
تاریخ: ۱۳۹۲/۰۴/۱۵ ۶:۷

سلام، از نظر شما متشکرم

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 لاگ 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 برین و کد زیر رو اضافه کنید:

```
<trust level="Full" originUrl=".*" />
```

2) باید تغییری رو در متدهای الحاقی Encrypt و Decrypt بدین، که از این متدها برای رمزنگاری و رمزگشایی محتوای کوکی تصویر امنیتی استفاده میشه. قبل از هر کدوم از این متدها flag زیر رو اضافه کنید:

```
[System.Security.Permissions.PermissionSet(System.Security.Permissions.SecurityAction.Assert,
Unrestricted = true)]
```

همچنین، یک خط داخل بدنه 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: ۱۳۹۲/۰۴/۱۵ ۹:۷

بله کار کرد. متشکرم.

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۲/۰۴/۲۹ ۱۸:۲۹

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: ۱۳۹۲/۰۵/۰۱ ۲۱:۵۱

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. لطفا کل عملیات از تشکیل تصویر امنیتی، تا مقداری که درون کوکی قرار می‌گیره، و بعد رمز گزاری میشه ، تا مقداری که درون کوکی مرورگر ذخیره میشه (برای مشاهده کوکی‌های مرورگر از این روش می‌تونید استفاده کنید) . سپس ببینید بعد از پست شدن فرم حاوی کپچا به سرور مقداری که از درون کوکی خونده میشه و می‌خواد decrypt بشه با مقداری که از اول ذخیره شده یکسان باشه.

امیدوارم مشکلتون حل شه دوست عزیز.

نویسنده: کوروش ت
تاریخ: ۱۳۹۲/۰۹/۲۸ ۱۰:۳

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: ۱۳۹۲/۰۹/۲۸ ۱۰:۹

ویژگی AllowAnonymous را بالای آن قرار دهید. همچنین Url.Action دو پارامتر نام کنترلر و نام اکشن متد را دریافت می‌کند (برای استفاده از آن در سایر Viewهایی که در کنترلر جاری نیستند) .

نویسنده: کوروش ت
تاریخ: ۱۳۹۲/۱۰/۱۵ ۹:۱۳

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 میکنم، تصاویر نمایش داده نمی‌شود. ممنون میشم راهنمایی کنید.

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۱۵ ۹:۵۷

برای دیباگ کردن این مساله در سمت سرور بهتر است از ELMAH استفاده کنید. لاگ آن‌را بررسی کنید شاید جایی در پشت صحنه خطایی صادر می‌شود.

نویسنده: کوروش ت
تاریخ: ۱۳۹۲/۱۰/۱۵ ۱۰:۲۷

با سلام، Elmah Log رو چک کردم، مشکل دسترسی داشت که در قسمت Web.config تگ Trust رو اضافه کردم و فعلاً ظاهر شد. اضافه کردن trust مشکل امنیتی ایجاد نمی‌کنه؟

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۱۵ ۱۰:۳۶

کدوم قسمتش مشکل دسترسی داشت؟ چه خطایی گرفتید دقیقا؟ چه trust ایی رو اضافه کردید؟ (در هاستی که بتونید trust level رو تغییر بدید، یعنی عملا از لحاظ امنیتی درست تنظیم نشده و مداخل رو قفل نکرده)

نویسنده: کوروش ت
تاریخ: ۱۳۹۲/۱۰/۱۵ ۱۰:۴۸

پیغام:

```
The system cannot find the file specified " source=" mscorlib "
```

کد تراست

```
< trust level = "Medium" originUrl = ".*" />
```

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۱۵ ۱۱:۳۷

عجیبه که کار کرده. چون با حالت medium مشکل داره .

نویسنده: کوروش ت
تاریخ: ۱۳۹۲/۱۰/۱۵ ۱۳:۰

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: ۱۳۹۲/۱۰/۱۵ ۱۴:۵۶

trust level یعنی کد شما مثلا 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 می‌تونید کدهای آن را بررسی کنید.

نویسنده: کوروش ت
تاریخ: ۱۳۹۲/۱۰/۱۵ ۱۵:۲۶

منظور من این نبود، منظورم خطری برای سایر Third Party APIها بود که دادن تراست لول خطری برای ای پی آی‌های دیگه نداشته باشه. در مورد این کد هیچ چیز مبهمی وجود نداره و هم توضیحات خوب داده شده بود و هم از متدهای رادست و قابل فهم استفاده شده بود. از راهنمایی‌های 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 خطایی نمیگیرم و تصویر امنیتی بدرستی نمایش داده می‌شود ولی روی پورت 8080 تصویر کپچا لود نمیشود و خطای زیر را در کنسول کروم لاگ 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 می‌خواستم 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 نمی‌شود و اصلاً کنترولر CaptchaImage فراخوانی نمی‌شود ممنون میشم راهنمایی کنید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۰۷ ۱۱:۴۶

اگر فراخوانی نمی‌شود، 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 برید به صفحه جدید (مثلا بعد لاگین برید به صفحه مدیر) دوباره back بزنی همون مقدار رو می‌بینید و متاسفانه 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 CaptchaImageResult.cs نگارش دوم 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);
```

متد الحاقی DisableBrowserCache در مطلب غیرفعال کردن کش مرورگر در MVC ارائه شده.

2- امضای متد Index کنترل Home نیاز به NoBrowserCache دارد (کل صفحه‌ی لاگین کش نشود):

```
public class HomeController : Controller
    {
        [NoBrowserCache]
        public ActionResult Index()
```

3- امضای متد CaptchaImage را در کنترلر Home 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:

```
<img src="@Url.Action("CaptchaImage", "Home", routeValues: new{ rdnDate = DateTime.Now.Ticks })"/>
```

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 integrated به classic می‌بردی و از network service برای identity

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 به web.config نبود.

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 requestهای احتمالی از همانجا انجام می‌شد. در هر صورت بابت این کد مفید تشکر می‌کنم.

نویسنده: محسن درپرستی
تاریخ: ۱۳۹۳/۰۹/۰۱ ۱۴:۴۵

یک مورد دیگه که باعث بروز این مشکل میشه افزونه فایرباگ هست. علتش رو نمی‌دونم اما در زمان فعال بودن فایرباگ دو بار درخواست برای تصویر امنیتی ارسال و مقدار درون کوکی با تصویری که می‌بینید متفاوت میشه.
درحالی که در زمان خاموش بودن فایرباگ این مشکل وجود نداره.

نویسنده: امیر نوروزیان
تاریخ: ۱۳۹۳/۱۰/۲۱ ۹:۱۴

سلام؛ من از این روش استفاده می‌کنم. هر از چند وقت گیر میده که خطایی اتفاق افتاده؛ تریس میکنم، این خطا رو نشون میده:

```
{"Error occurred while decoding OAEP padding."}
```

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۳/۱۱/۰۴ ۶:۱۸

سلام
من تمامی مراحل طی کردم اما متاسفانه روی سرور با اینکه فول تراست هست همین خطا دریافت میکنم

نویسنده: پژمان پارسائی
تاریخ: ۱۳۹۳/۱۱/۰۷ ۹:۳۵

با سلام خدمت کلیه دوستان و همکاران گرامی
از اینکه نمی‌تونم جواب نظرات رو بدم عذرخواهی می‌کنم. (به دلیل خدمت سربازی :دی)
بنا به پیشنهاد جناب نصیری و دیگر دوستان ی ک مخزن کد Github برای این تصویر امنیتی درست کردم. اصلاح باگ‌های برطرف شده اون هم تا اینجا طبق نظرات همین مطلب اعمال شدن. هر کی می‌خواد تو این پروژه مشارکت داشته باشه PR بفرسته .

نویسنده: مهدی سعیدی فر
تاریخ: ۱۳۹۴/۰۱/۰۸ ۱۰:۳۹

اگر بخواهم که کاربر با وارد کردن مستقیم آدرس ایجاد کپچا، کد آن اجرا نشود اضافه کردن کد زیر به ابتدای متد ExecuteResult کلاس CaptchaImageResult درست هست؟
البته در این کد Null reference می‌گیرم ولی خب ایجاد کپچا متوقف میشود. راه دیگه ای هست که بدون Null refrence جواب بدهد؟

```
if (context.RequestContext.HttpContext.Request.UrlReferrer.AbsolutePath ==
                context.RequestContext.HttpContext.Request.Url.AbsolutePath)
                throw new InvalidOperationException();
```

نویسنده: محسن خان

تاریخ: ۱۳۹۴/۰۱/۰۸ ۱۳:۵۹

اگر ActionResult سفارشی هست، از context.HttpContext.Response.StatusCode = 403 و بعد context.HttpContext.Response.End استفاده کنید. اگر ActionFilterAttribute سفارشی تعریف شده، از filterContext.Result = new Http403Result کمک بگیرید.

نویسنده: محسن درپرستی
تاریخ: ۱۳۹۴/۰۱/۲۳ ۱۸:۳

بعد از انتقال از یک سرور به سرور دیگر ، خطایی در زمان نمایش کپچا ایجاد می‌شود :

```
System.Security.Cryptography.CryptographicException: Object already exists at
System.Security.Cryptography.CryptographicException.ThrowCryptographicException(Int32 hr)   at
System.Security.Cryptography.Utils._CreateCSP(CspParameters param, Boolean randomKeyContainer,
SafeProvHandle&amp; hProv)
```

متن کامل خطا :

captcha-error.txt

تنها با دادن دسترسی کامل به Everyone برای مسیر زیر مشکل برطرف شد:

```
C:\ProgramData\Microsoft\Crypto\RSA\MachineKeys
```

در حالی که نه در سیستم لوکال و نه در سرور قبلی این مشکل وجود نداشت. سرور جدید 2012 هست.
استفاده از Full Trust و یا دستور زیر هم کارساز نشد :

```
aspnet_regiis -pa "SampleKeys" "NT AUTHORITY\NETWORK SERVICE"
```

عنوان: الگوی PRG در ASP.NET MVC
نویسنده: محسن اسماعیل پور
تاریخ: ۱۳۹۲/۰۳/۰۶ ۱۱:۰
آدرس: www.dotnettips.info
برچسب‌ها: Design patterns, MVC, ASP.Net MVC

تا حالا با این پنجره حتما مواجه شدین:

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 بدونن 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 متد GET به کاربر نشون داده بشه، کاربر Redirect بشه به صفحه. برای توضیح این مسئله به سراغ AccountController که بصورت پش 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. مثلا معمولا پس از انجام عمل ثبت نام، کاربر به صفحه شخصی یا صفحه اصلی سایت منتقل میشه و یا در موقع ویرایش اطلاعات پش 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 مروگر اعلام میکنه 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 303 نمی‌بینم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۰۷ ۱۳:۳۲

البته روش توضیح داده شده همان روش متداول PRG است. اگر حتما نیاز به 303 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 TestMvcRPG.Helper;

namespace TestMvcPRG.Controllers
{
    public class HomeController : BaseController
    {
        [HttpGet]
        public ActionResult Index()
        {
            return View();
        }
```

```
        [HttpPost]
        public ActionResult Index(string data)
        {
            if (ModelState.IsValid)
            {
                return Redirect303("Index"); // post-redirect-get
            }
            return View();
        }
    }
}
```

عنوان: **اعتبارسنجی سایتهای چند زبانه در ASP.NET MVC - قسمت اول**
نویسنده: محسن اسماعیل پور
تاریخ: ۱۳۹۲/۰۳/۰۸ ۰:۵
آدرس: www.dotnettips.info
برچسب‌ها: ASP.Net MVC, jQuery, Validation, Localization, Globalization, MVC

اگر در حال تهیه یک سایت چند زبانه هستید و همچنین سری مقالات Globalization در ASP.NET MVC رو دنبال کرده باشید میدانید که با تغییر Culture فایلهای Resource مورد نظر بارگذاری و نوشته‌های 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 هست:

```
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);
}
```

Culture 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 رو اضافه کنید:

```
@using (Html.BeginForm())
{
    <ol>
        <li>
            @Html.LabelFor(m => m.DecimalValue)
            @Html.TextBoxFor(m => m.DecimalValue)
            @Html.ValidationMessageFor(m => m.DecimalValue)
        </li>
```

```
        <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 میشوید که مقادیر وارد شده نامعتبر میباشند.

اگر پروژه رو در حالت دیباگ اجرا کنیم و نگاهی به داخل ModelState بیاندازیم، میبینیم که کاراکتر جدا کننده قسمت اعشاری برای fa-IR '/' میباشد که در اینجا برای اعداد مورد نظر کاراکتر '.' وارد شده است.

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 jQuery 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. درداخل 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.cshtm بروید و فایلهای جاوا 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 globalize.culture.fa-IR.js کاراکتر جدا کننده اعشاری '.' در نظر گرفته شده است که مجبور به تغییر آن هستیم. برای اینکار فایل را باز کرده و numberFormat را پیدا کنید 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 مینمایم و در ادامه نحوه اعتبارسنجی را در unobtrusive 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 (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 خطا صادر شود.

برای اعتبارسنجی تاریخ شمسی متاسفانه 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 کردام 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 کاملا اصولی نباشد ولی به هر حال در اینجا توضیح میدهم. در فایل 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: ["شنبه", "جمعه", "پنجشنبه", "چهارشنبه", "سه شنبه", "دوشنبه", "یکشنبه"],
        namesAbbr: ["شنبه", "جمعه", "پنجشنبه", "چهارشنبه", "سه شنبه", "دوشنبه", "یکشنبه"],
        namesShort: ["ش", "ج", "پ", "چ", "س", "د", "ی"]
    },
    months: {
        names: ["ژانویه", "فوریه", "مارس", "آوریل", "می", "ژوئن", "ژوئیه", "اوت", "سپتامبر", "اکتبر", "نوامبر", "دسامبر", "", ""],
        namesAbbr: ["ژانویه", "فوریه", "مارس", "آوریل", "می", "ژوئن", "ژوئیه", "اوت", "سپتامبر", "اکتبر", "نوامبر", "دسامبر", "", ""]
    },
    AM: ["ق.ظ", "ق.ظ", "ق.ظ"],
    PM: ["ب.ظ", "ب.ظ", "ب.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, 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 31/12/1392 وارد شده باشد و در ادامه برای بررسی بیشتر تاریخ یک بار به میلادی و تاریخ میلادی دوباره به شمسی 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:

۱ - yyyy/MM/dd

۲ - dd/MM/yyyy

البته از توابع اعتبارسنجی تاریخ میتوانید به صورت جدا استفاده نمایید و لزومی ندارد آنها را همراه با jQuery Globalize بکار ببرید. در آخر خروجی کار به این شکل است:

در کل استفاده از jQuery Globalize برای اعتبارسنجی در سایتهای 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.

عنوان: آغاز به کار با Twitter Bootstrap در ASP.NET MVC
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۱۴ ۱۲:۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, CSS Framework, Twitter Bootstrap

Twitter Bootstrap یک فریم ورک 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 سایت‌هایی زیبا، مانند دکمه‌ها، عناصر فرم‌ها، منوها، ویجت‌ها و غیره است. تمام این‌ها نیز همانطور که عنوان شد برمبنای HTML، CSS و جاوا اسکریپت تهیه شده‌اند؛ بنابراین در هر نوع فناوری سمت سروری مانند ASP.NET، PHP، روبی 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 نمائید 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 هستند که اصطلاحا 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 js 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 ASP.NET MVC

امکانات Bundling and minification در نوع پروژه‌های نسبتا جامع‌تر ASP.NET MVC به صورت پیش فرض لحاظ شده است. اما اگر یک پروژه خالی را شروع کرده‌اید، نیاز است بسته نیوگت آن‌را نیز نصب کنید:

```
PM> Install-Package Microsoft.AspNet.Web.Optimization
```

پس از آن، کلاس کمکی BundleConfig ذیل را به پروژه جاری اضافه نمائید. 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 مباحث Bundling and minification در حالت Debug 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=vsUQD0OJg4AJ-RZH8jSRRCu_rjl2U1nZrmSsaUyxoAc1" 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 Succes، Error، Info و امثال آن. این نام‌ها مفهومی را می‌رسانند؛ اما در مورد نحوه پیاده سازی آن‌ها جزئیاتی را بیان نمی‌کنند.
برای نمونه می‌توان کلاسی را به نام redText ایجاد کرد. هر چند این نام، توضیحاتی را در مورد علت وجودی‌اش 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 مثلا شخصی، به آن رنگ ویژه علاقه دارد، یا اینکه قرمز است بخاطر نمایش خطایی 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 جدیدی برای 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.
Semantic class names 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 hover، color و width برگرفته شده از Twitter Bootstrap را انتساب داد و نهایتا به نتیجه دلخواه رسید؛ بدون اینکه نگران باشیم افزودن کلاس جدیدی در اینجا بر روی سایر کلاس‌های انتساب داده شده، تاثیر منفی دارد.

ج) Conventions

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 شکیل ظاهر خواهد شد. یا برای مثال اگر به یک table html کلاس‌های table table-striped table-hover را انتساب دهیم، به صورت خودکار قراردادهای پیش فرض table 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.

طرحبندی صفحات یک سایت به کمک Twitter Bootstrap

بررسی Grid layouts

Layout به معنای طرحبندی و چیدمان محتوا در یک صفحه است. یکی از متداولترین روش‌های طرحبندی صفحات چه در حالت چاپی و چه در صفحات وب، چیدمان مبتنی بر جداول و گریدها 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. Grid در طرحبندی، 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، عموما کارهای طراحی صفحات به کمک HTML Tables انجام می‌شد. اما با پخته‌تر شدن CSS، استفاده از Tables برای طراحی صفحات کمتر و کمتر گشت تا اینکه نهایتا فریم ورک‌های CSS ایی پدید آمدند تا طراحی‌های مبتنی بر CSS را با ارائه گریدها، ساده‌تر کنند. مانند 960 GS، Blue print و ... Twitter Bootstrap که طراحی مبتنی بر گریدهای CSS ایی 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 Text Text Text Text Text Text Text Text Text Text Text Text Text 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 خواهند گرفت. اما قصد داریم این محتوا را، دو ستونی نمایش دهیم. Div به همراه Title 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 Text Text Text Text Text Text Text Text Text Text Text Text Text 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 span 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 Viewایی 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 class="span1">1</div>
        <div class="span1">1</div>
        <div class="span1">1</div>
        <div class="span1">1</div>
        <div class="span1">1</div>
        <div class="span1">1</div>
        <div class="span1">1</div>
        <div class="span1">1</div>
        <div class="span1">1</div>
        <div class="span1">1</div>
        <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 حداکثر 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 ثابتش نمایش 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 ایی 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

حالت Fluid Grid، عرض هر span برحسب درصد تعیین می‌شود. 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 خواهند کرد؛ اما در حالت Fixed Grid خیر.

حالت Fluid، شیوه متداول استفاده از bootstrap در اکثر سایت‌های 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 زوم 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 px480 و یا کمتر، تا px1200 و یا بیشتر را پوشش می‌دهد.

به این ترتیب با اندازه px940 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-tablet و visible-desktop به علاوه hidden-phone، hidden-tablet و hidden-desktop. 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 الزاما 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>
    </div>
```

نظرات خوانندگان

نویسنده: احمدعلی شفیعی
تاریخ: ۱۳۹۲/۰۳/۱۹ ۱۰:۸

محدودیت ۱۲تایی ستون‌ها به اندازه‌ی صفحه بستگی داره. برای من ۱۳تاست.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۱۹ ۱۷:۳۵

شعار این کتابخانه چنین چیزی هست: « Bootstrap is built on responsive 12-column grids ». یعنی حالت پیش فرض طراحی آن 12 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 تعریف شده

```
.row-fluid .span12 {
  width: 100%;
  *width: 99.94680851063829%;
}
```

container هم با عرض 940 پیکسل تعریف شده. همچنین یک GutterWidth را هم باید در بین ستون‌ها درنظر داشت. حالت پیش فرض:

```
@gridColumns:             12;
@gridColumnWidth:         60px;
@gridGutterWidth:         20px;
```

البته میشه مثلا عرض رو بیشتر از 940 تعریف کرد. برای اینکار باید css رو ویرایش و از ابزارهایی مانند less استفاده کرد و یا امکان سفارشی سازی اون هم هست.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۲۰ ۱۲:۴۸

یک نکته تکمیلی

در مطلب فوق یک چنین تعریفی 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 آن‌را نصب کنید).
در این بین، اگر فایل bootstrap.css را اندکی ویرایش کنید (مثلا نام فونت‌ها یا اندازه آن‌ها را تغییر دهید)، متوجه خواهید شد که تغییرات شما اصلا 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 غیرفشرده 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 جزو استاندارد آن است. در ASP.NET 4.0 می‌شود از امکانات ASP.NET MVC 4 هم استفاده کرد (همان 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 css 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-l3czPS8KW6usEOpK8aIglVqPOVEKWoSXV4N-5N2Q1"></script>
```

منتها برای CSS فقط فایل Style.css و برای جاوااسکریپت فقط jQuery رو اضافه میکنه، به نظرتون مشکل چی میتونه باشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۵/۱۸ ۲۱:۷

اگر واقعا مشکلی وجود داشته باشد، در همین لینک‌های پویایی که ارائه دادید، با کلیک بر روی آن‌ها عبارت «Minification failed. Returning unminified contents» 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 برروی لینک‌های پویا تولید شده به صورت زیر است :

مثلا با کلیک بر روی css تولید شده فایل Style.css را مشاهده میکنم :

و با کلیک بر روی فایل js تولید شده نیز به صورت زیر می‌باشد :

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۵/۱۹ ۰:۵۸

چند سؤال اینجا هست که از راه دور قابل بررسی نیست:
- چطور تعاریف فایل‌ها رو در کدها اضافه کردید؟ (اگر خطای نبود jQuery رو بدون خطای کامپایل که ذکر شد گرفتید، یعنی اصلا jQuery تعریف نشده در تعریف bundle نهایی)
- ساختار پوشه‌های 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 پیوست شده که شامل فایل‌های اسکریپت و css متناظر با هر بحث می‌شود. این فایل‌ها در این سری ویژه، مورد تائید هستند و آزمایش شده‌اند.

نویسنده: فرزان
تاریخ: ۱۳۹۲/۰۶/۱۸ ۱۱:۱۲

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>)'
```

مواجه میشم. در واقع از کلاس AsIsBundleOrderer ارور میگیره.
ممنون میشم راهنمایی کنید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۶/۱۸ ۱۱:۲۷

- در نگارش جدید System.Web.Optimization ، امضای FileInfo متد OrderFiles تبدیل شده به BundleFile . (همین تغییر نام کافی است)
- در زمان نگارش این مطلب، FileInfo بوده.

نویسنده: احسان شیروان
تاریخ: ۱۳۹۲/۰۷/۰۴ ۱۷:۳۲

با سلام ضمن تشکر از شما بابت آموزش‌های بسیار خوب
یه سوال داشتم من از VS2010 , C#.net دارم استفاده می‌کنم برای استفاده کردن از Bootstrap تمامی مراحل یاد شده توسط شما رو مو به مو اجرا کردم اما متاسفانه به هیچ وجه کار نداد البته خطا هم نمیده اما مباحث مربوط به ریسپانسیو بودن صفحه و حتی نمایش spanها جوابی نگرفتم آیا این بدلیل ورژن VS2010 هست یا دات نت؟ یا یه چیز دیگه ؟
ظاهرا سایتی که ایجاد کردم هیچ درکی از span نداره در حالی که تمام فایل‌های bootstrap رو طبق آموزش روی سایت نصب کردم
ممنون میشم راهنماییم نمایید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۷/۰۴ 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 مباحث آن تغییر کرده‌اند. فایل‌های بوت استرپ 2 در قسمت‌های بعدی این بحث، پیوست 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 (قبلا درست کار می‌کرد) اما توی نسخه 4.5 کار نمی‌کنه، بعد از دیباگ فهمیدم اصلا فایلها با BundleCollection اضافه نمیشه و زمانی که می‌خواد خروجی بگیره با خطای نبود مدخل مورد نظر مواجه میشه. Update-Package را هم اجرا کردم اما همون خطا 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 بوت‌استرپ 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 PM> Install-Package

Microsoft.AspNet.Web.Optimization بسته‌ها رو نصب کردم و یک پوشه به اسم 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 بفرمائید؟ با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۰۳ ۲۳:۱۳

مطلب 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 افتادنی و یکسری چیزها دیگه کار نمیکنه.در ضمن پروژه خالی mvc شروع کردم.(تو حالت basic خودش پیش فرض داره)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۱۹ ۱۳:۲۸

کمی بالاتر پاسخ دادم «مطلب جاری و سری مرتبط با آن برای بوت استرپ **2** نوشته شد . .. ». (چندین بار)

نویسنده: ایمان باقری
تاریخ: ۱۳۹۳/۰۱/۱۹ ۱۳:۳۷

من هم بوت استرپ 2 رو گرفتم فقط فایل‌های جی کوئری و validation اون رو ندارم
فایل های jquery.validate.min.js jquery.unobtrusive-ajax.min.js jquery-1.9.1.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 MVC 5 را شروع کردید. پیش فرض استاندارد MVC 5 بوت استرپ 3 است و نه دو.
فایل‌های کامل بوت استرپ 2 را از طریق دستور ذیل که پیشتر نیز ذکر شده بود دریافت کنید (تمام اجزای js 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 پروژه mvc 4 رو شروع کردم(بالا هم عرض کردم) و بوت استرپ هم از سایت rbootstrap.ir نسخه v 2.3.2 رو گرفتم.
البته الان که از دستور PM> Install-Package Twitter.BootstrapRTL -Version 2.3.2.1 استفاده کردم jquery-1.9.1.min.js رو هم نصب کرد

و مشکل منوها حل شد.فقط فایل‌های validation و ajax رو 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: (^ و ^)

نویسنده: M.Shakeri
تاریخ: ۱۳۹۳/۰۶/۲۱ ۱۴:۲۱

شما اشاره کردید به این که اگر پروژه رو درحالت release قرار بدیم فشرده سازی اعمال میشه یعنی اینکه در فایل web.config در قسمت compilation خصوصیت debug رو 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: ۱۳۹۳/۰۶/۲۱ ۱۴:۴۲

پس از بررسی مجدد، مشکلی مشاهده نشد:
- رندر حالت فشرده شده در حالت غیر دیباگ:

- رندر حالت معمولی و اصل آن در حالت دیباگ:

در کل اگر این روش برای شما کار نمی‌کند، شرط HttpContext.Current.IsDebuggingEnabled را از متد addBundle حذف کنید و else آن‌را نگه دارید.

نویسنده: امیرتنها
تاریخ: ۱۳۹۳/۱۰/۰۵ ۲۳:۱۷

با سلام

هنگام اجرا این خطا رو میده .

'does not contain a definition for 'AddStyles

```
Line 19: @BundleConfig.AddStyles("~/Content/css",


 Line 20: "~/Content/bootstrap.css",
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۰۵ ۲۳:۴۵

یکبار نظرات این بحث را کامل مطالعه کنید. این مطلب برای بوت استرپ 2 نوشته شد، در آن زمان. نگارش 3 آن در اینجا بحث شده. همچنین این کلاس‌ها تغییراتی داشته‌اند که پیشتر در نظرات بحث شده.

عنوان: **استفاده از Twitter Bootstrap در کارهای روزمره طراحی وب**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۱۵ ۲۱:۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, CSS Framework, Twitter Bootstrap

پس از آشنایی مقدماتی با Twitter Bootstrap، در این قسمت قصد داریم تا با ویژگی‌هایی از آن آشنا شویم که در کارهای رومزه 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 جهت دعوت خواننده به مطالعه مطالب، بجای فراری دادن او، در مباحث تایپوگرافی رعایت می‌شوند. خوشبختانه Twitter Bootstrap به همراه یک سری تنظیمات تایپوگرافی پیش فرض است که در ادامه آن‌ها را مرور خواهیم کرد.

پیش فرض‌های ابتدایی آن‌را 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>
```

سرتیتر 1
سرتیتر 2
سرتیتر 3
سرتیتر 4
سرتیتر 5
سرتیتر 6

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 strong، small و em نیز اعمال می‌کند:

```
<div class="row-fluid">
            <div class="span12">
                <p class="lead">
                    تیتر</p>
```

```
            <p>
                متن متن متن
```

در bootstrap 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 citeبرای ذکر ماخذ نقل قول ذکر شده:

```
<div class="row-fluid">
            <div class="span12">
                <blockquote>
                    <p>
                        جهت نمایش نقل قول
                    </p>
                    <small><cite title="کاربر
شماره 2"></cite> از شخصی</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. یك
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 definition 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>
```

```
                    <dd>
                        توضیحات بلند در اینجا</dd>
                </dl>
            </div>
        </div>
```

عنوان توضیحات بلند در اینجا

در bootstrap 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 دهید. 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>
```

inline code

code
code
code

استفاده از جداول و تاثیر bootstrap بر آن‌ها

در ادامه کدهای یک جدول متداول را که مزین شده‌است 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>
                            متن سه 2
                        </td>
                    </tr>
                    <tr>
                        <td>
                            متن یک 3
                        </td>
                        <td>
                            متن دو 3
                        </td>
                        <td>
                            متن سه 3
                        </td>
                    </tr>
                </tbody>
            </table>
        </div>
    </div>
</div>
```

عنواني خاص در اينجا

ستون يك	ستون دو	ستون سه
1 متن يك	1 متن دو	1 متن سه
2 متن يك	2 متن دو	2 متن سه
3 متن يك	3 متن دو	3 متن سه

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 گوشه‌های 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 شده‌است تا اثر آن‌را 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" />
```

```
                    <label>
                        نام:</label>
                    <input name="name" type="text"
/>
                </fieldset>
                </form>
            </div>
        </div>
    </div>
```

در اینجا یک فرم ساده را مشاهده می‌کنید. بدون bootstrap، تمام عناصر این فرم در یک سطر نمایش داده می‌شوند. اما با صرفا 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 داده‌است 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 مرورگر، یعنی قرار دادن تمام عناصر در پشت سر هم را فعال کنید، تنها کافی است کلاس form-inline را به form تعریف شده فوق اضافه نمائید.

2) نکاتی در مورد checkboxes و radio buttons

در حالت پیش فرض، با تعریف یک label و checkbox، برچسب متناظر 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 radio استفاده خواهد شد):

```
<label class="checkbox">
       <input type="checkbox" name="isMale" />
 مذکر</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 یا radio button در یک سطر (با توجه به حالت چیدمان عمودی پیش فرض فرم‌ها)، ابتدا آن‌ها را داخل یک div قرار دهید. سپس به تمام checkboxها یا radio buttonها کلاس inline را نیز اضافه نمائید. برای مثال:

```
<div>
                        <label class="radio
inline">
                            <input type="radio"
name="isMale" />
                            مذکر</label>
                        <label class="radio
inline">
                            <input type="radio"
name="isFemale" />
                            مؤنث</label>
                    </div>
```

4) تعیین اندازه فیلدها

با استفاده از کلاس‌هایی مانند input-mini، input-small، input-medium، input-large، input-xlarge و input-xxlarge 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>
```

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 دکمه نیز btn درنظر 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 دیده‌است.

```
<form class="form-horizontal" action="/signup"
method="post">
                <fieldset>
```

```
                    <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" />
                                مذکر</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 ایی با کلاس control-group محصور نمائید.
- به برچسب‌ها، کلاس control-label را انتساب دهید.
- کنترل‌های مدنظر را در div ایی 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">لطفا ایمیل را با
فرمت صحیحی وارد نمائید</span>
     </div>
</div>
```

برای تزریق خطاهای اعتبارسنجی ویژه، در اینجا می‌توان از کلاس‌های help-block و یا help-inline به همراه کلاس‌هایی مانند warning، info، error و success 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 و spanهایی با کلاس add-on، می‌توان عناصری بصری خاصی را به عنوان جزئی از فیلد مورد نظر اضافه کرد.

9) HTML Helpers مخصوص ASP.NET MVC برای کار با bootstrap

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://mvc4bootstaphelper.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 تزئین 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>
                <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>
                        <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>
</div>
```

دکمه	لینک	کلاس بکار گرفته شده
default	default	btn
primary	primary	btn btn-primary
info	info	btn btn-info
success	success	btn btn-success
warning	warning	btn btn-warning
danger	danger	btn btn-danger
inverse	inverse	btn btn-inverse
link	link	btn btn-link
large	large	btn btn-primary btn-large
small	small	btn btn-primary btn-small
mini	mini	btn btn-primary btn-mini
block	block	btn btn-primary btn-block
disabled	disabled	btn btn-primary disabled

همانطور که ملاحظه کردید، الزامی ندارد که این کلاس‌ها را حتما به دکمه‌ها اعمال کرد. برای نمونه می‌توان از یک span یا لینک نیز برای تعریف دکمه‌ها بهره جست. برای یکپارچه سازی چنین دکمه‌هایی (که در اصل دکمه نیستند) با ASP.NET MVC 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 گوشه‌های آن، نمایش دایره‌ای و یا نمایش به همراه حاشیه یک تصویر خواهند شد.
Twitter Bootstrap به همراه صدها آیکون ارائه شده است. این آیکون‌ها توسط glyphicons.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 پیش فرض 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:

```
<div class="input-prepend">
        <span class="add-on"><i class="icon-
envelope"></i></span>
        <input type="email" dir="ltr" />
    </div>
```

نظرات خوانندگان

نویسنده: pezhman
تاریخ: ۱۳۹۲/۰۳/۱۸ ۱۵:۲

سلام
ممنون بابت آموزش
می خواستم بدونم چطور از راست به چپ رو درست کردین؟ مخصوصا فرم‌ها رو
آیا از pull-right استفاده کردین یا css دیگه؟
ممنون :)

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 آموزشهاتون
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 نمیان. (واقع گرایی)
- جواب کلی: یک سری شبیه ساز رو از اینجا می‌تونید دریافت کنید. امکان یکپارچه سازی با VS.NET هم وجود دارد.

عنوان: نگاهی به اجزای تعاملی Twitter Bootstrap
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۱۸ ۸:۵
آدرس: www.dotnettips.info
برچسب‌ها: MVC, CSS Framework, Twitter Bootstrap

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) در Twitter Bootstrap

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
                    <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>
                        <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 متداول تعریف منو را در Bootstrap، ملاحظه می‌کنید. 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، تعریف شده است.

المان‌های منو، توسط لیست‌ها تعریف می‌شوند. این لیست باید با کلاس dropdown-menu شروع شود. سپس هر عضو این لیست، یک آیتم منو را تشکیل خوهد داد. اگر نیاز به خط جدا کننده‌ای در این میان باشد، باید از کلاس divider استفاده کرد.

در اینجا همچنین امکان تعریف 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 می‌شود. زیر منو نیز دارای کلاس dropdown-menu خواهد بود.

چند نکته تکمیلی در مورد منوها:

- همانطور که در قسمت بررسی دکمه‌های Bootstrap نیز عنوان شد، برای تبدیل یک المان به دکمه فقط کافی است کلاس btn را به آن انتساب دهیم. در اینجا نیز می‌توان کلاس لینک منوی پایین افتادنی را به صورت btn btn-primary dropdown-toggle تعریف کرد تا ظاهر آن تبدیل به یک دکمه Bootstrap شود.

- در اولین div مثال فوق، کلاس dropdown ذکر شده است. می‌توان این کلاس را به dropup تغییر داد تا نحوه باز شدن منوی تعریف 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 می‌دهد، 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"
                        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 نکته‌ای که در اینجا نسبت به حالت قبل اضافه شده است، استفاده از کلاس btn-group است. این کلاس سبب می‌شود تا دکمه‌های تعریف شده در کنار هم به شکل یک toolbar مرتب شوند.

تعریف برگه‌ها در Twitter Bootstrap

برای تعریف منوهای راهبری سایت به شکل برگه، کار با استفاده از کلاس nav و تعریف یک لیست ساده شروع می‌شود:

```
<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>
```

- 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 گوشه‌های گرد استفاده می‌کند.
- اگر می‌خواهید این گوشه‌ها گرد نباشند، از nav-list استفاده کنید.
- در هر کدام از حالت‌ها می‌توان از یک گزینه اختیاری با کلاس nav-header برای 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.

تعریف 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 ... data ...
                        </div>
                        <div class="tab-pane" id="tab2">
                            data2 ... data2 ...
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
```

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 hrefهای تعریف شده به tab1 و tab2 اشاره می‌کنند. ایندو در حقیقت id محتوای متناظر هستند که در قسمت div ایی با کلاس tab-content به div هایی با id مساوی 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 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>
                </nav>
            </div>
        </div>
    </div>
```

navbar نیز همانند بسیاری از کامپوننت‌های Bootstrap، بر اساس ul و liها کار می‌کند. در ابتدای کار، کلاس nav را به تگ ul انتساب می‌دهیم. این مورد با مباحث 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 navbar-inner را اضافه خواهیم کرد.

عموما 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 margin-top مرتبط با 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 قرارگیرند 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 منویی مخفی است را نمایش دهیم می‌توان از کلاس btn-navbar استفاده کرد؛ که نمونه‌ای از آن‌را 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 یك / سطح دو / سطح سه

عموما 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 آن‌را 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 امثال آن که در مثال ذیل ذکر شده‌اند، رنگ‌های متفاوتی را سبب خواهند شد. برای گرد کردن بیش از حد گوشه 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">اخطار</span></li>
                        <li><span class="label label-importatnt">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-importatnt">مهم است</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>
</div>
```

پیش فرض
موفقیت آمیز
اخطار
مهم است
اطلاعات
رنگ معکوس

پیش فرض
موفقیت آمیز
اخطار
مهم 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 alert-danger، رنگ‌های متفاوتی را ارائه خواهند داد و اگر علاقمندید که کاربر بتواند این اخطار را با کلیک بر روی دکمه ضربدر کنار آن حذف کند، از یک دکمه با کلاس close و data-dismiss مساوی alert 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 روش کلیک می‌کنم هیچ اتفاقی نمی‌افته و باز نمی‌شه.
CSSها JSها رو هم در باندل و هم مستقیما در layout اضافه کردم باز جواب نگرفتم
update-package هم هیچ کمکی نکرد لطفا راهنمایی کنید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۲۹ ۱۹:۳۹

دریافت یک نمونه مثال: BootstrapMenu.zip
+ نحوه بازسازی وابستگی‌ها

عنوان: **اعمال کلاس‌های ویژه اعتبارسنجی Twitter bootstrap به فرم‌های ASP.NET MVC**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۲۰ ۸:۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, Twitter Bootstrap

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("", "لطفا مشکلات را برطرف کنید!");
                    this.ModelState.AddModelError("Name", "نام فقط باید وحید باشد!");
                    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 = "تعریف 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>تعریف 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 متد 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 divایی با کلاس controls محصور شده‌اند.

7) قسمت دکمه فرم، در div ایی با کلاس form-actions قرار گرفته تا یک زمینه خاکستری در اینجا ظاهر شود.

8) دکمه فرم، با کلاس btn خاص bootstrap تزئین شده.

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.findByName(element.name).addClass(errorClass).removeClass(validClass);
                } else {
                    $(element).addClass(errorClass).removeClass(validClass);
                    $(element).closest('.control-group').removeClass('success').addClass('error');
                }
                $(element).trigger('highlated');
            },
            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('unhighlated');
            }
        });

        $(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 bootstrap است. همچنین در حالت postback و نمایش خطاهای سفارشی، قسمت بررسی field-validation-error 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:

و یا اگر کاربر فیلد را تکمیل کند، رنگ فیلد و ردیف تعیین اعتبار شده، سبز می‌شود:

و در حالت خطاهای سفارشی سمت سرور، پس از postback، شکل زیر نمایش داده می‌شود:

- لطفا مشکلات را برطرف کنید!

تعریف کاربر

نام: بررسي — نام فقط باید وحید باشد!

نام خانوادگي: 2

ارسال

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۰۵ ۱۵:۵

دریافت مثال این قسمت: BootstrapValidation.zip
+ نحوه بازسازی وابستگی‌ها

نویسنده: حمید چگینی
تاریخ: ۱۳۹۲/۰۹/۲۵ ۱۴:۱۹

ممکنه در مورد validator.setDefaults.$ بیشتر توضیح بدین. طبق مستندات تنظیمات پیش فرض اصلاح میشه.
اما متوجه این مطلب نمی‌شم. دو تابع highlight و unhighlight 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: ۱۳۹۲/۰۹/۲۵ ۱۴:۳۳

بله. وجود دارند. لیست کامل آن‌ها را در مستندات آن مطالعه کنید.

عنوان: **ویرایش قالب پیش فرض Add View در ASP.NET MVC برای سازگار سازی آن با Twitter bootstrap**
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 این‌کارها به صورت خودکار توسط VS.NET بجای قالب پیش فرض ایجاد 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 X\Common7\IDE\ItemTemplates\CSharp\Web\MVC X\CodeTemplates قرار دارند. برای نمونه اگر بخواهیم پیش فرض‌های تولید فرم‌های MVC4 را تغییر دهیم، باید به پوشه MVC 4\CodeTemplates\AddView\CSHTML 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 CodeTemplates\AddView\CSHTML به همراه تمام فایل‌های tt آن، در پوشه جاری پروژه مدنظر ما کپی شود. به این ترتیب ابتدا به این پوشه 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 پوشه CodeTemplates\AddView\CSHTML\Create.tt ریشه پروژه جاری قرار دارد، اعمال کنیم.
الف) ابتدا نام این فایل را به CreateBootstrapForm.tt تغییر می‌دهیم. از این لحاظ که این نام جدید در drop down مرتبط با scaffold template صفحه Add view 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 آن‌را 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 nested properties یا complex 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

و یا اگر نیاز به اطلاعات بیشتری داشتید، پروژه MvcScaffolding یک چنین روابطی رو پشتیبانی می‌کند: « MvcScaffolding: One-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: ۱۳۹۳/۰۴/۱۴ ۱۶:۲

سلام. من از vs2013 update 2 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 تمپلیتم رو نمیبینم. ساختار فولدر بندی در ریشه پروژم وجود داره CodeTemplates\AddView\CSHTML\MyCreate.tt

نویسنده: محمد آزاد
تاریخ: ۱۳۹۳/۰۴/۱۴ ۱۷:۱۳

جواب رو پیدا کردم توی mvc 5 ساختار فولدرها متفاوت هستش http://weblogs.asp.net/imranbaloch/customizing-the-asp-net-mvc-5-web-api-2-scaffolding-templates

عنوان: استفاده از افزونه Typeahead مجموعه Twitter Bootstrap در ASP.NET MVC
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۲۲ ۸:۰
آدرس: www.dotnettips.info
برچسب‌ها: MVC, Twitter Bootstrap

با تعدادی از کامپوننت‌های Bootstrap در مطلب « نگاهی به اجزای تعاملی Twitter Bootstrap » آشنا شدید. یکی دیگر از این افزونه‌ها، Typeahead نام دارد که در حقیقت نوعی Autocomplete text box است. در ادامه قصد داریم نحوه استفاده از آن‌را در ASP.NET MVC بررسی کنیم.

استفاده‌ی استاتیک 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 رشته JSON معادل آن شده و توسط ViewBag.JsonString به View ارسال می‌شود.
View 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 حداکثر 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
                              })
```

به این ترتیب، یک چنین خروجی در صفحه درج می‌شود:

```
<input autocomplete="off" data-items="8" data-provide="typeahead"
data-source="[&quot;Afghanistan&quot;,&quot;Albania&quot;,&quot;Algeria&quot;,&quot;American
Samoa&quot;,&quot;Andorra&quot;,&quot;Angola&quot;,&quot;Anguilla&quot;,&quot;Antarctica&quot;,&quot;An
tigua and/or Barbuda&quot;]"
id="search" name="search" type="text" value="" />
```

همانطور که ملاحظه می‌کنید دقیقا data-source تهیه شده مطابق نیاز خاص این افزونه، html encoded است. به علاوه هر جایی در htmlAttributes صفحه از under line 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 ایی 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 متدی که یک عبارت، یا قسمتی از آن را از طریق پارامتر term دریافت و خروجی JSON مناسبی را ارائه می‌کند، همانند متد 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 متد getJSON کتابخانه jQuery، مقدار عبارت وارد شده در TextBox جستجو، به آدرس اکشن متد GetNames ارسال و سپس حاصل به source افزونه typeahead انتساب داده می‌شود.

نظرات خوانندگان

نویسنده: امیر هاشم زاده
تاریخ: ۱۳۹۲/۰۶/۲۵ ۱۰:۴۵

آیا پیاده سازی تابع matcher افزونه typeahead با استفاده از Htmlhelper امکان پذیر 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

matcher یک callback 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
        //  this.query کوئری جاری را بر می‌گرداند.
        return true // اگر آیتم تطابق داشته است
    }
})
```

مثلا:

```
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 خطایی 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

مطلب 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
1.چطور میشه نتایج را با یک قالب خاص نمایش داد (Template) نمایش تصویر ، گروهبندی و ... برای هر مورد پیشنهادی
2.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

IRIS از Chosen استفاده کرده؛ سایت جاری از jQuery Autocomplete .

نویسنده: رضا
تاریخ: ۱۳۹۲/۰۹/۱۹ ۱۰:۱۵

ممنون 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 highlighter آن باید 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 مثلا 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.
یک مثال کامل: Heavily Customizing a Bootstrap Typeahead

عنوان: **استفاده از modal dialogs مجموعه Twitter Bootstrap برای گرفتن تائید 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 آن‌را در دست می‌گیرند و تا زمانیکه این صفحه ویژه بسته نشود، امکان استفاده از صفحه زیرین، وجود نخواهد 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 کپسوله کنیم، می‌توان یک افزونه جدید جی‌کوئری را با نام فایل jquery.bootstrap-modal-confirm.js برای این منظور تدارک دید:

```
// <![CDATA[
(function ($) {
    $.bootstrapModalConfirm = function (options) {
        var defaults = {
            caption: 'تائید عملیات',
            body: 'آیا عملیات درخواستی اجرا شود؟',
            onConfirm: null,
            confirmText: 'تائید',
            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>' +
                   '<div class="modal-body">'
                        + options.body + '</div>' +
                   '<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 دو دکمه تائید 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 تائید کلیک کرد، یک callback به نام onConfirm 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(); //می‌خواهیم لینک به صورت معمول عمل نکند
                $.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 تائید، قسمت onConfirm اجرا خواهد شد.

Index
حذف رکورد
تائید عملیات
آیا عملیات درخواستی اجرا شود؟
تائید
انصراف

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 می‌تونیم تو یه لیست ازش استفاده کنیم ..یعنی هریک از این حذف‌ها تو یه سطر باشن

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۳۱ ۱۳:۱

مطلب + نظرات « حذف یک ردیف از اطلاعات به همراه پویانمایی محو شدن اطلاعات آن توسط jQuery در ASP.NET MVC »

نویسنده: محمدرضا2
تاریخ: ۱۳۹۳/۰۶/۰۲ ۱۵:۵۶

سلام.

چطور می‌توان سایز Modal را Dynamic 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: ^

عنوان: **Bundling and Minifying Inline Css and Js**
نویسنده: م کریمی
تاریخ: ۱۳۹۲/۰۳/۲۳ ۱۷:۵
آدرس: www.dotnettips.info
برچسب‌ها: MVC, MS Ajax Minifier, Performance

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 Asp.Net MVC 4 بهینه نماییم .
در ابتدا با استفاده از Nuget پکیج BundleMinifyInlineJsCss را به پروژه MVC خود مطابق شکل زیر اضافه می‌نماییم .

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

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

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

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

در ادامه پروژه را اجرا می‌کنیم. 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 باشه اصلا نیازی به حذف فواصل خالی در HTML نهایی نیست. چون حداقل کاری رو که الگوریتم‌های فشرده سازی خوب انجام می‌دن، مدیریت فضاهای خالی است.

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

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

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

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 ، مرسی از مطالب خوبتون
بحث Optimization در لینک زیر کاملتر بررسی شده، البته مطلب شما برایم تازگی داشت.
http://go.microsoft.com/fwlink/?LinkId=254725

نکاتی که میشه گفت
در وب فرم هم قابل استفاده است
برای single download کردن فایل‌های css و js دو روش وجود دارد:
1. تنظیم debug=false در بخش compilation در فایل Web.config
2. نوشتن کد زیر در کلاسی که باندل‌های خود را در bundleTable درج می‌کنید.

```
BundleTable.EnableOptimizations = true
```

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

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

که با نصب آن، 4 کتابخانه به referenceهای پروژه اضافه می‌شود.
تفاوتی که در scriptها ایجاد می‌کند می‌توان به حذف کردن description ها، تغییر در variableها ، و min کردن jsهای شما اشاره کرد.
موفق باشید.

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

این موضوع به بیان‌های مختلفی در سایت تابحال مطرح شده

نحوه ارتقاء برنامه‌های موجود MVC3 به MVC4
قابلیت‌های کاربردی ASP.NET WebFroms -
اسکریپت‌های خود را یکی کنید
فشرده سازی فایل‌های CSS و JavaScript بصورت خودکار توسط MS Ajax Minifier

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

با سلام
مباحثی همچون Bundling and Minifying قبلا توسط آقای نصیری توضیح کامل داده شده است . ما هم مباحث تکمیلی را اضافه می‌کنیم

عنوان: **نمایش فرم‌های مودال Ajax ایی در ASP.NET MVC به کمک Twitter Bootstrap**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۳/۲۴ ۸:۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, Twitter Bootstrap

اصول نمایش اطلاعات مودال به کمک bootstrap در مطلب « استفاده از modal dialogs مجموعه Twitter Bootstrap برای گرفتن تائید از کاربر » بررسی شدند.
در این قسمت قصد داریم یک فرم Ajaxایی را در ASP.NET MVC به همراه تمام مسایل اعتبارسنجی، پردازش اطلاعات و غیره را به کمک Twitter Bootstrap و jQuery Ajax پیاده سازی کنیم.

تهیه افزونه jquery.bootstrap-modal-ajax-form.js

از این جهت که مباحث مرتبط با نمایش و پردازش فرم‌های مودال Ajaxایی 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 مثلا 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>";
        enableBootstrapStyleValidation(); //اعمال نکات خاص بوت استرپ جهت اعتبارسنجی یکپارچه با آن
        $.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) {
                    window.location = options.loginUrl; //در حالت لاگین نبودن شخص اجرا می‌شود
                }
                else if (status === 'error' || !data) {
                    if (options.errorHandler)
                        options.errorHandler();
                }
                else {
                    var dialogContainer = "#dialogDiv";
                    $(dialogContainer).remove();
                    $(mainContainer).appendTo('body');

                    $('#dialogContent').html(data); // دریافت پویای اطلاعات مودال دیالوگ
                    $.validator.unobtrusive.parse("#dialogContent"); // فعال سازی اعتبارسنجی فرمی که با
ایجکس بارگذاری شده
                    enablePostbackValidation();
                    // و سپس نمایش آن به صورت مودال
                    $('#dialogDiv').modal({
                        backdrop: 'static', //با کلیک کاربر روی صفحه، صفحه مودال بسته نمی‌شود
                        keyboard: true
                    }, 'show');
                    // تحت نظر قرار دادن این فرم اضافه شده
                    processAjaxForm('#dialogContent');
                }
            }
        });
    };
})(jQuery);
// ]]>
```

توضیحات:

- توابع enableBootstrapStyleValidation و enablePostbackValidation در مطلب « اعمال کلاس‌های ویژه اعتبارسنجی Twitter bootstrap به فرم‌های ASP.NET MVC » 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 Ajaxایی دریافتی، مسایل اعتبارسنجی را به آن اعمال کرده و سپس دکمه submit آن‌را 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 Ajaxایی 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 امن‌تر 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 });
            }

            this.ModelState.AddModelError("", "خطایی رخ داده است");
```

```
            return PartialView("_ModalPartialView", user);
        }
    }
}
```

کدهای کنترلر برنامه در این حالت از سه قسمت تشکیل می‌شود:

الف) متد Index حالت HttpGet که صفحه ابتدایی را نمایش خواهد داد.

ب) متد RenderModalPartialView یک partial view اضافه شده به برنامه به نام _ModalPartialView.cshtml را بازگشت می‌دهد. این partial view در حقیقت همان فرمی است که قرار است به صورت مودال نمایش داده شود و پردازش آن نیز Ajaxایی باشد.

ج) متد Index حالت HttpPost که نهایتا اطلاعات فرم مودال را دریافت خواهد کرد. اگر پردازش موفقیت آمیز بود، نیاز است همانند کدهای فوق return Json صورت گیرد. در غیراینصورت مجددا 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: برای حالتیکه نیاز به به روز رسانی کامل صفحه 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 رندر 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 form-horizontal است. همچنین divهای modal-body، modal-header 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
افزودن کاربر جدید
مشخصات کاربر
نام
لطفا نام را تکمیل کنید
نام خانوادگي
لطفا نام خانوادگي را تکمیل کنید
ارسال
انصراف

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۰۵ ۱۴:۵۶

دریافت مثال این قسمت: BootstrapPopupAjaxForm.zip
+ نحوه بازسازی وابستگی‌ها

نویسنده: دانشجو
تاریخ: ۱۳۹۲/۰۴/۱۸ ۲۳:۲۳

عرض پاپ آپ چطور تعیین میشه؟
امکان تمام صفحه شدن آن وجود دارد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۱۸ ۲۳:۳۷

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', //با کلیک کاربر روی صفحه، صفحه مودال بسته نمی‌شود
                        keyboard: true
                    }, 'show').css({
                                'width': '100px',
                                'max-width': '1000px',
                                'margin-left': '10px' });
```

نویسنده: علی رضا امیدیان
تاریخ: ۱۳۹۲/۰۴/۲۷ ۱۸:۲۹

با سلام
در اکشن زیر

```
publicActionResult Index(User user) //ذخیره سازی اطلاعات
{
   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: ۱۳۹۲/۰۴/۲۷ ۱۹:۵۸

در متد processAjaxForm در قسمت options.errorHandler آن چون dialogContent مجددا مقدار دهی می‌شود، نیاز است سیم کشی‌های آن مجددا 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: ۱۳۹۲/۰۴/۲۸ ۱۴:۳۷

با سلام و تشکر از پاسخ شما

نویسنده: پیام
تاریخ: ۱۳۹۲/۰۶/۰۱ ۸:۵۵

با سلام آقای نصیری من برای استفاده از این مثال تو سایت خودم مجبور از بسته jquery-1.9.1 استفاده کنم این درحالی که من برای اجرای منوی superfish باید از بسته ای jquery-1.6 به پایین استفاده کنم برای حل همچین مشکلاتی چه کاری باید کنم ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۶/۰۱ ۹:۰

- از آخرین نسخه افزونه یاد شده استفاده کنید. اگر ندارد، با نویسنده آن تماس بگیرید و درخواست ارتقاء آن‌را بدهید.
- jquery-1.9.1 نباید تداخلی با کار شما داشته باشد. یک سری از مسایل از نگارش 2 آن به بعد هست که حذف شدند و منسوخ شده اعلام گردیدند. برای رفع آن ، نیاز است از کتابخانه jQuery Migrate نیز استفاده کنید.

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۲/۰۷/۱۰ ۲۰:۳۴

ممنون از شما واقعاً مطلب جالبی بود،
فقط یک سوال :
کلاس‌های مدل من Product و Category است، نحوه نمایش گروه محصولات در DropDownList به چه صورتی است؟
ابتدا من یک ViewModel ایجاد کردم :

```
public class ProductCategoryViewModel
{
public Product Product{get;set;}
public Category Category{get;set;}
}
```

و سپس در RenderModalPartialView به PartialView مربوطه پاس دادم :

```
public ActionResult RenderModalPartialView()
        {
            //رندر پارشال ویوو صفحه مودال به همراه اطلاعات مورد نیاز آن
            return PartialView(viewName: "_ModalPartialView", model: new ProductCategoryViewModel());
        }
```

در PartialView هم برای نمایش نام گروه محصولات از طریق ViewBag لیست گروه محصولات داده شده را در یک

DropDownList به صورت زیر استفاده کردم :

```
@Html.DropDownListFor(model => model.Product.CategoryId, (SelectList)ViewBag.CategryNames, "--گروه 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 مدال بازنویسی کنیم؟ البته ظاهراً باید توی قسمت :

```
if(mvcHost.IsPartialView) {
#>

<#
```

قالب موردنظر (_ModalPartialView) برای حالت مدال فرم رو باید بنویسم، من با سینتکس T4 آشنا نیستم ، میشه لطف کنید راهنمایی کنید؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۷/۱۰ ۲۱:۵۳

- این مطلب برای **بوت استرپ 2** نوشته شده. (از فایل‌های نگارش 3 استفاده نکنید)
+ چه چیزی رو نمایش نمی‌ده؟ 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 متد RenderModalPartialView که نوشتید ViewBag ایی مقدار دهی نشده. بنابراین دیتاسورس مدنظر شما نال هست.
روش بهتر این است که یک خاصیت به ViewModel ایی که تعریف کردید اضافه کنید:

```
public IEnumerable<SelectListItem> Names { get; set; }
```

بعد این خاصیت را در زمان بازگشت اطلاعات از اکشن متد مقدار 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 این‌دو با هم اشتباه گرفته نشوند یک _ در ابتدای نام partial view قرار می‌دهند. فقط چند div مخصوص modal- header و موارد دیگری که در متن ذکر شده را باید به قالب قبلی اضافه کنید تا برای فرم‌های مودال هم کار کند. تفاوت دیگری ندارد.

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۲/۰۷/۱۰ ۲۱:۵۵

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: ۱۳۹۲/۰۹/۰۲ ۵:۳۸

حال یه سوال اگه بخواهیم مثلا 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 پنچره مودال یه پیام مثلا به شکل پاپ آپ به کاربر نشون بدیم مبنی بر صحت انجام کار باید به چه صورت عمل کرد یا مثل بالا جایی که شما نوشتین خطایی رخ داده است. منظورم شبیه چیزیه که تو همین وبلاگ عمل میشه یعنی اینجا پس از مثلا نوشتن این نظر و ارسال بدون اشکال به سرور به بنده یه پیام نشون میده که بدونم نتیجه‌ی کارمو
با تشکر :)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۰۲ ۹:۲۶

این مورد پیش بینی شده در این مثال. اگر به خروجی اکشن متد دقت کنید، چنین چیزی است:

```
return Json(new { success = true });
```

همین مورد یعنی بررسی مقدار success دریافتی، جایی که عملیات Ajax ایی ارسال اطلاعات پایان می‌یابد انجام شده:

```
if (result.success) {
                            $('#dialogDiv').modal('hide');
                            if (options.completeHandler)
                                options.completeHandler();
```

بنابراین تنها کاری که باید انجام بدید، قرار دادن کدهای نمایش اطلاعات نهایی در callbackهای completeHandler ویا errorHandler مربوط به افزونه bootstrapModalAjaxForm.$ 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 صدها افزونه jQuery موجود برای نمایش پیام‌ها.

نویسنده: مسعود بهرامی
تاریخ: ۱۳۹۲/۰۹/۰۲ ۱۶:۵۰

مرسی؛ یه مورد دیگه با عرض پوزش تو این فرم مودال چون Ajax هستش یه مشکل که داریم وقتیه که بخواهیم دیتا سورس رو مثلا برای کومبو 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 اصلا کومبو 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 اونو دستی بایند کردم به ViewBag اما باز هم قایده نداشت چون کومبو باکس کندو اصلا Render نمیشه ممنون میشم اگه راهنمایی کنین :)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۰۲ ۱۹:۲

کمی بالاتر در جواب سؤال مشابهی پاسخ دادم. باید یک ViewModel درست کنید جایی که return PartialView توسط عملیات Ajax ایی درخواست می‌شود، منبع داده View در حال رندر را تامین کند.

نویسنده: مجید
تاریخ: ۱۳۹۲/۰۹/۱۴ ۲۲:۰

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 undefine 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 متد 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 مدال ارسال کنیم و در آنجا نمایش دهیم؟ «این مورد پیش بینی شده در این مثال. اگر به خروجی اکشن متد دقت کنید، چنین چیزی است:

return Json(new { success = true });	1

همین مورد یعنی بررسی مقدار success دریافتی، جایی که عملیات Ajax ایی ارسال اطلاعات پایان می‌یابد انجام شده:

if (result.success) {	1
$('#dialogDiv').modal('hide');	2
if (options.completeHandler)	3
options.completeHandler();	4

بنابراین تنها کاری که باید انجام بدید، قرار دادن کدهای نمایش اطلاعات نهایی در callbackهای completeHandler ویا errorHandler مربوط به افزونه $.bootstrapModalAjaxForm است. اینجا دست شما باز است. اگر علاقمند بودید از یک alert ساده استفاده کنید»

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۲۰ ۲۳:۵۸

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 نهایتا در سمت کلاینت تحت عنوان result دریافت شده. یا مستقیما با آن کار کنید یا این result را به متد completeHandler ارسال کنید تا در جای دیگری توسط completeHandler تعریف شده قابل استفاده باشد.

نویسنده: ع طاهری
تاریخ: ۱۳۹۲/۱۲/۲۵ ۱۴:۶

سلام.
برای پیغامهایی که در این سایت نمایش داده می‌شوند مانند "پیغامهای خطا" به رنگ قرمز یا "پیغامهای info" به رنگ آبی و ...که در وسط صفحه ظاهر شده و بعد از چند ثانیه ناپدید می‌شوند از modal استفاده شد؟ لطفا راهنماییم کنید که چطور میتونم همچین پیغامهایی را نمایش دهم.
با سپاس فراوان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۲۵ ۱۴:۲۴

از noty استفاده شده.

نویسنده: مجتبی فخاری
تاریخ: ۱۳۹۳/۰۲/۰۱ ۰:۳۵

با سلام
اگه بخواهیم برای Edit از این مدال استفاده کنیم چطوری می‌توانیم اطلاعات مثلا Id را به متد مورد نظر پاس بدهیم. آیا باید در این خط renderModalPartialViewData: {}, Id را ارسال نماییم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۰۱ ۰:۴۴

```
$.ajax({
            type: "POST",
            url: options.renderModalPartialViewUrl,
```

```
            data: options.renderModalPartialViewData,
```

مطابق سورس فوق، data ارسالی عملیات ajax با renderModalPartialViewData مقدار دهی می‌شود.

نویسنده: M.Shakeri
تاریخ: ۱۳۹۳/۰۵/۲۵ ۱۴:۵۷

تو مثالی که آپلود کردین اگه پارامتر بفرستیم خطای 500 میده.

```
$.bootstrapModalAjaxForm({
                    postUrl: '@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: ۱۳۹۳/۰۵/۲۵ ۱۵:۳۶

```
renderModalPartialViewData:  JSON.stringify({ id: 1 }),
```

عنوان: MVC vs 3-Tier Pattern
نویسنده: مهرداد اشکانی
تاریخ: ۱۳۹۲/۰۳/۲۶ ۱۴:۰
آدرس: www.dotnettips.info
گروه‌ها: Design patterns, MVC, Architecture, معماری سه لایه, MVC معماری

من تا به حال برنامه نویس‌های زیادی را دیده‌ام که می‌پرسند «چه تفاوتی بین الگوهای معماری MVC و Three-Tier وجود دارد؟» قصد 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 مربوطه‌اشان هستند.

Three-Tier

MVC

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 Business logic به لایه Data 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 لایه ها.

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 آورده ام.

نظرات خوانندگان

نویسنده: محسن اسماعیل پور
تاریخ: ۱۳۹۲/۰۳/۲۶ ۲۲:۳۳

3-Layer در واقع Architecture Style هست اما MVC یک Design Pattern هست پس مقایسه مستقیم نمیدونم کاری دست باشد یا نه اما میتونیم به این شکل نتیجه گیری کنیم:

Data Access: شامل کلاسهای ADO.NET یا EF برای کار با دیتابیس.

Business Logic: یا همان Domain logic که میتوان Model رو به عنوان Business entity در این لایه بکار برد.

UI Layer: بکارگیری Controller و View در این لایه

نویسنده: یزدان
تاریخ: ۱۳۹۲/۰۳/۲۷ ۱:۳۳

در برنامه نویسی 3لایه کار Business Logic به طور واضح و شفاف چی هست و چه کارهایی در این لایه لحاظ میشه ؟

نویسنده: fss
تاریخ: ۱۳۹۲/۰۳/۲۷ ۸:۴۱

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 میتونید لایه UI رو با یکی از روش ها، مثلا سیلورلایت، asp.net mvc یا asp.net web form پیاده کنید.

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۳/۲۷ ۸:۴۹

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 نمی‌تونید MVC و Web forms رو در یک سطح قرار بدی (که اگر اینطور بود اصلا نیازی به MVC نبود؛ نیازی به MVVM برای سیلورلایت یا WPF نبود و یا نیازی به MVP برای WinForms یا Web forms نبود).

نویسنده: fss
تاریخ: ۱۳۹۲/۰۳/۲۷ ۹:۱۳

دوست عزیز من متوجه منظور شما نشدم. حرف من اینه که MVC، MVVM، MVP و .. در سطح UI پیاده میشن.

نویسنده: مهرداد اشکانی
تاریخ: ۱۳۹۲/۰۳/۲۷ ۹:۱۸

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: ۱۳۹۲/۰۳/۲۷ ۹:۳۸

این طور نیست دوست عزیز شما می‌تونید 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: ۱۳۹۲/۰۳/۲۷ ۱۷:۱۴

به بنظر بنده هم معماری رو نمی‌شود با الگو مقایسه کرد به هر حال خود الگوی mvcها یک سری لایه داره و تا اونجایی هم که می‌دونم فرق tier با layer اینه که tierها رو از لحاظ فیزیکی هم جدا می‌کنند

نویسنده: وحید فرهمندیان
تاریخ: ۱۳۹۳/۱۲/۰۴ ۱۷:۱۶

لایه کسب و کار مغز برنامه شما میباشد. یک زمانی میخواهید معادله ریاضی حل کنید در این لایه و زمانی نیز نیاز است مقداری داده از انباره داده خود بخوانید. لذا UI درخواست محاسبه معادله یا استخراج گزارش را به کسب و کار میدهد، کسب و کار بررسی میکند تا درخواست را پاسخ دهد. اگر برای پاسخ نیاز به انباره داده بود به لایه داده میفرستد تا مطابق با آن درخواست داده‌های مناسب استخراج شده و برگشت داده شوند.
نکته ای که وجود دارد این است که لایه داده حتما نباید با یک پایگاه داده ارتباط برقرار کند، و لایه UI نیز نباید شخصا کار پردازشی یا منطقی انجام دهد و این کارها باید به لایه کسب و کار ارجاع داده شوند.

عنوان: **پلاگین جستجو با jquery و twitter bootstrap**
نویسنده: ایاک
تاریخ: ۱۳۹۲/۰۳/۲۹ ۲۱:۳۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, jQuery-Plugins, Widget factory, Twitter Bootstrap

در این مطلب با نحوه استفاده از پلاگین جستجوی سفارشی searchboxmvc.js آشنا خواهید شد.

قبلاً در اینجا با نحوه ایجاد پلاگین jQuey آشنا شدید. روشی دیگری نیز برای ایجاد این نوع پلاگین‌ها وجود دارد و آن استفاده از widget factory موجود در پلاگین jQuery UI 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 =
"startWith")
        {
            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 = "Utppq-981", FirstName = "علی", LastName = "دایی" });
            lst.Add(new Person() { Id = 5, Code = "zttsn-471", FirstName = "علی", LastName = "کریمی"
});
```

```
            lst.Add(new Person() { Id = 6, Code = "poiud-901", FirstName = "مهدی", LastName = "مهدوی
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: "startWith",
            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

loadUrl : آدرس اکشن متدی است که بصورت ajax ای فراخوانی شده و نتایج حاصل را بازگشت میدهد.
نتایج حاصله باید با فرمت json بازگشت داده شوند. اگر نتایج موفقیت باشد باید بصورت Json(new { Status = "OK",}) Records = models بازگشت داده شوند و اگر خطایی در این بین صورت گرفت مقدار Status نباید مقدار OK باشد.
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.
defaultStringFilterMode : اگر فیلد مورد جستجو از نوع رشته ای باشد (یعنی isStringType آن برابر true باشد) آنگاه پارامتر سوم اکشن متد بطور خودکار مقداردهی خواهد شد. مقادیر این خاصیت میتواند startWith یا contains و یا equal باشد.
loadDataOnLeave : اگر برابر false باشد، هربار که متن input تغییر کرد بلافاصله یک تقاضا برای یافتن مقادیر به سرور فرستاده

میشود و نیازی نیست که فوکوس از کنترل خارج شود.

displayClass : نام کلاس css است که به div 3 اعمال خواهد شد.

displayNoResultClass : در صورتیکه جستجو نتیجه ای نداشته باشد این کلاس به div 3 اعمال خواهد شد.

display : یک فانکشن که برای ایجاد خروجی html برای نمایش در div 3 بکار می‌رود.

listItemsDisplay : یک فانکشن که برای ایجاد خروجی html برای آیتم‌ها بکار می‌رود.

fields : یک آرایه از فیلدهای موردنیاز پلاگین .

خاصیت‌های فیلد نیز بصورت زیر است:
fieldName : نام فیلد

fieldTitle : عنوان فیلد
defaultValueField : فیلد پیش فرض که جستجو بر اساس آن صورت می‌گیرد. اگر تعیین نشود فیلد اول آرایه به عنوان فیلد پیش فرض انتخاب خواهد شد.

defaultDisplayField : فیلد پیش فرض که برای نمایش متن div 3 بکار می‌رود(البته اگر پارامتر display تعیین نشود)

filter : اگر برابر true باشد این فیلد در لیست فیلدهای جستجو خواهد آمد و کاربر می‌تواند بر اساس آن جستجو انجام دهد.

isStringType : اگر برابر true باشد ، پارامتر سوم اکشن متد 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 bootstrap به فرم‌های ASP.NET MVC » می‌باشد. بنابراین تعاریف مدل و کنترلر آن به همراه توضیحات ذکر شده در آن، در ادامه مورد استفاده قرار خواهند 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 تعریف 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.
placement، 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.

تبدیل خطاهای اعتبارسنجی ASP.NET MVC به Tooltip

هدف ما در اینجا، نهایتا رسیدن به شکل زیر می‌باشد:

همانطور که ملاحظه می‌کنید، اینبار بجای نمایش خطاها در یک برچسب مقابل کنترل متناظر، این خطا صرفا 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
                $("." + this.settings.validClass).tooltip("destroy");
                //افزودن تول‌تیپ‌ها
                for (var i = 0; i < errorList.length; i++) {
                    var error = errorList[i];
                    $(error.element).tooltip({ trigger: "focus" }) // فقط در حالت فوکوس نمایش داده
شود
                                .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 » می‌باشند و از تکرار مجدد آن‌ها در اینجا صرفنظر گردید.

توضیحات

- با توجه به اینکه دیگر نمی‌خواهیم خطاها به صورت برچسب در مقابل کنترل‌ها نمایش داده شوند، کلیه Html.ValidationMessageFor به صورت کامنت درآورده شده‌اند.

- تغییر دوم مطلب جاری، اضافه شدن متد showErrors به تنظیمات پیش فرض jQuery Validator است. در این متد، 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 متد tooltip بوت 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="لطفا نام را تکمیل کنید"
 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="لطفا نام را تکمیل کنید" id="Name" name="Name" type="text" value=""
data-original-title="لطفا نام را تکمیل کنید" title="">
```

این مقدار توسط افزونه tooltip بوت استرپ شناسایی شده و مورد استفاده قرار می‌گیرد.

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۰۵ ۱۴:۴۷

دریافت مثال این قسمت: BootstrapTooltip.zip
+ نحوه بازسازی وابستگی‌ها

نویسنده: مهدی پایروند
تاریخ: ۱۳۹۳/۰۲/۱۰ ۹:۱۸

روش متداول افزودن این tooltip به یک المان جدید که به صفحه اضافه شده چیست؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۱۰ ۹:۲۹

```
$('body').tooltip({
    selector: '[rel=tooltip]'
});
```

نویسنده: مهدی پایروند
تاریخ: ۱۳۹۳/۰۲/۱۰ ۹:۵۴

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: ۱۳۹۳/۰۹/۱۲ ۱۷:۴۵

با سلام
چرا بعد از عملیات ajax ایی 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: ۱۳۹۳/۰۹/۱۲ ۱۷:۵۷

- 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 ASP.NET MVC به شکل Popover به کمک Twitter bootstrap**
نویسنده: 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 bootstrap به فرم‌های ASP.NET MVC » می‌باشد. بنابراین تعاریف مدل و کنترلر آن، به همراه توضیحات ذکر شده در آن، در ادامه مورد استفاده قرار خواهند 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. نهایتا 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 مجددا بازیابی نشده و مکان اسکرول عمودی صفحه، تغییر نکند.

تبدیل خطاهای اعتبارسنجی ASP.NET MVC به PopOver

هدف ما در اینجا نهایتا رسیدن به شکل زیر می‌باشد:

همانطور که ملاحظه می‌کنید، اینبار بجای نمایش خطاها 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 معتبر است نیاز به نمایش پاپ اور ندارد
                $("." + this.settings.validClass).popover("destroy");
                //افزودن پاپ اورها
                for (var i = 0; i < errorList.length; i++) {
                    var error = errorList[i];
```

```
                    $(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 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 » می‌باشند و از تکرار مجدد آن‌ها در اینجا صرفنظر گردید.

توضیحات

- با توجه به اینکه دیگر نمی‌خواهیم خطاها به صورت برچسب در مقابل کنترل‌ها نمایش داده شوند، کلیه Html.ValidationMessageFor به صورت کامنت درآورده شده‌اند.

- تغییر دوم مطلب جاری، اضافه شدن متد showErrors به تنظیمات پیش فرض jQuery Validator است. در این متد، 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 متد popover بوت 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="لطفا نام را تکمیل کنید"
 id="Name" name="Name" type="text" value="" />
```

اما در اینجا به صورت پویا، data-original-title و data-content نیز به آن افزوده می‌گردند:

```
<input class="text-box single-line input-validation-error" data-val="true" data-val-required="لطفا نام را
تکمیل کنید"
 id="Name" name="Name" type="text" value=""
data-original-title="خطای اعتبارسنجی" title="" data-content="لطفا نام را تکمیل کنید">
```

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: ۱۳۹۲/۰۴/۰۵ ۱۴:۴۱

دریافت مثال این قسمت: BootstrapPopover.zip

+ نحوه بازسازی وابستگی‌ها

عنوان: **ساخت قالب‌های نمایشی و ادیتور دکمه سه وضعیتی سازگار با Twitter bootstrap در ASP.NET MVC**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۰۴ ۸:۰
آدرس: www.dotnettips.info
برچسب‌ها: MVC, Twitter Bootstrap

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 محصور 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:

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 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 nullable bool است. البته مشکلی هم با نوع‌های bool معمولی ندارد. در حالت nullable، 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 واقعا کاربر اطلاعاتی را انتخاب کرده یا اینکه بدون توجه به فیلدها، 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 کلاس‌های btn-group و ویژگی data-toggle آشنا شدید. دقیقا این سه دکمه نیز

در اینجا به همین نحو تعریف شده‌اند.

در ابتدای نمایش یک View، خصوصا 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 آن‌را در سه متغیر noIsSelected، yesIsSelected و isIndeterminate ابتدای تعاریف قالب مشاهده می‌کنید.

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 آن‌را به این فیلد که نهایتا به سرور ارسال خواهد شد، اعمال خواهیم کرد.

تهیه قالب نمایشی Views\Shared\DisplayTemplates\BootstrapBoolean.cshtml

```
@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 صرفا نمایشی، فایل قالب BootstrapBoolean.cshtml قرار گرفته در مسیر Views\Shared\DisplayTemplates، یک چنین تعاریفی را خواهد داشت.

و نهایتا 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 گزینه ها، 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

data-toggle=buttons-radio ذکر شده در مقدمه بحث، زمانی عمل می‌کنه (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:

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.

تهیه OpenSearchResult سفارشی

برنامه باید بتواند محتوای XML ایی 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.WriteElementString("ShortName", ShortName);
                xmlWriter.WriteElementString("Description", Description);
                xmlWriter.WriteElementString("InputEncoding", "UTF-8");
                xmlWriter.WriteElementString("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 ایی مطابق نمونه‌ای است که در ابتدای توضیحات ملاحظه نمودید. توضیحات خواص آن، در ادامه مطلب ارائه شده‌اند.

تهیه 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) Url.Action ایی که در اینجا استفاده شده است مطابق تعاریف T4MVC است؛ ولی کلیات آن با نمونه پیش فرض ASP.NET MVC تفاوتی نمی‌کند. توسط T4MVC بجای ذکر نام اکشن متد 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 OpenSearchController به Header سایت

```
<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 پروژه و افزودن link خاص فوق به آن است. در این لینک، href آن باید به مسیر کنترلر OpenSearchایی که در قسمت قبل تعریف کردیم، اشاره کند. این مسیر نیز باید مطلق باشد. به همین جهت پارامتر protocol آن مقدار دهی شده است.

عنوان: Best Practice ی برای تأیید اعتبار کردن کاربران در ASP.NET MVC 4
نویسنده: محمد صاحب
تاریخ: ۱۳۹۲/۰۴/۱۰ ۱۶:۱۰
آدرس: www.dotnettips.info
گروه‌ها: Security, Authentication, ASP.Net MVC 4, MVC

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 رو به کلاس FilterConfig اضافه کنید.

```
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: ۱۳۹۲/۰۷/۰۳ ۹:۲۳

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: ۱۳۹۲/۰۷/۰۳ ۹:۴۲

«بعد از داخل اون Redirect میشه» خودکار هست یا دستی؟ اگر خودکار هست، خوب طبیعی هست جایی که AllowAnonymous نداره (یعنی اکشن متد بعدی 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: ۱۳۹۲/۰۷/۰۳ ۱۰:۲

اره خب حرفه شما درسته . اما هر دوتا اکشن من AllowAnonymous هست . در واقع از یک اکشن AllowAnonymous میخوام برم به یه اکشن AllowAnonymous داخل یه ناحیه . مشکل اینجاست که نمیره و میره تو صفحه لاگین .

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۷/۰۳ ۱۰:۱۹

ممکن هست در اون صفحه دوم داری چیزی رو نمایش می‌دی که ارجاعی داره به یک اکشن متد محافظت شده. مثلا صفحه‌ات چند قسمتی هست. یک قسمت داره از یک اکشن متد غیر عمومی شده اطلاعات دریافت می‌کنه.

نویسنده: ایمان
تاریخ: ۱۳۹۳/۰۴/۰۳ ۱۴:۳۴

سوال من اینه که اگه بخواهیم این محدودیت رو محدودتر کنیم[مثلا فقط نقش abc] دسترسی داشته باشه، اونوقت باید ["Authorize(Roles="abc")] رو روی اکشن بگذاریم؟

نویسنده: محمد صاحب
تاریخ: ۱۳۹۳/۰۴/۰۳ ۲۲:۶

اگه قراره روی فقط رو یه اکشن اعمال بشه درسته میتونی حتی میتونی روی کنترلر بزاری که تمام اکشن‌های اون کنترلر رو محدود کنه و اگه میخوای با روشی که تو این پست اومده عمل کنی باید خودت یک Authorize Attribute سفارشی بنویسی و به فیلترهای سراسری ت ادد کنی.

عنوان: **هدایت درخواست فایل‌های استاتیک در ASP.NET MVC به یک کنترلر**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۱۱ ۸:۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, Security

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 IIS 6 از موتور ASP.NET رد نخواهند شد. مگر اینکه این نوع پسوندها به aspnet_isapi.dll انتساب داده شوند. در IIS 7 به بعد این وضع بهبود یافته است. اگر تنظیم runAllManagedModulesForAllRequests در وب کانفیگ برنامه به true تنظیم شده باشد و برنامه در حالت Integrated pipeline بجای Classic mode اجرا شود، امکان مدیریت فایل‌های استاتیک نیز در برنامه‌های 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:
RouteCollection در ASP.NET MVC به 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 واقعا فایل‌های استاتیک قرار گرفته در پوشه Export به این کنترلر هدایت می‌شود یا خیر، یک breakpoint را بر روی سطر اول اکشن متد 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 متد Index کنترلر Export، نگاشت نخواهد شد (index در این مسیر ذکر نشده است).
برای حل این مشکل فقط کافی است مسیر یابی متناظری 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 متدی که 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             {                                   id "test.pdf"
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، عملا سایت از کار خواهد افتاد.

برای رفع این مشکل، در ابتدای متد RegisterRoutes فوق، تنظیمات ذیل را اضافه کنید تا پوشه‌های scripts، content و همچنین یک سری فایل با پسوند مشخص، همانند سابق و مستقیما 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 فولدر 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:

```
6   public class ExportController : Controller
7   {
8       public ActionResult Index(string id)
9       {
10          if (string.IsNullOrWhiteSpace(id))
11          {
12              return Redirect("/");
13          }
```

id "folder1/test.pdf"

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 فولدرهای 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}" );
```

ممنون

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 راسا 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 اصلا اجرا نمیشه. در متد registerRoutes همه آدرس رو تعریف کرده ام و routeexistingfiles رو هم true کردم یه کنترلر و یه متد هم 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 شرطی‌که 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.
متد 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، مباحث آن‌را 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 routes.RouteExistingFiles = true; و تنظیم مسیردهی RouteConfig اما با صدا زدن http://localhost/export/test جواب می‌گیرم ولی با http://localhost/expor/test.jpg خطای 404 دریافت می‌کنم

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

فرقی نمی‌کند: Mvc5RouteExistingFiles.zip
فقط اگر قرار است تمام فایل‌های استاتیک پوشه‌ی export، منهای 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

فایلهایی که به این صورت دانلود میشن قابلیت resume ندارن ، راهی برای resume وجود داره ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۲۱ ۲۳:۲۹

« File ActionResult هایی با قابلیت از سرگیری مجدد برای ASP.NET MVC »

نویسنده: احمد
تاریخ: ۱۳۹۳/۰۷/۱۶ ۱۳:۳۹

آیا این کار تاثیر زیادی توی سرعت کار میزاره؟
من میخوام واسه‌ی همه‌ی فایل‌های تصویری اینکارو بکنم، پیشنهاد میشه؟

یا مثلا با استفاده از این لینک

نویسنده: احمد
تاریخ: ۱۳۹۳/۰۷/۱۶ ۱۴:۴

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!

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۱۶ ۱۴:۲۱

این لینک در ASP.NET MVC یک چنین امضایی دارد؛ با خروجی return File:

```
public class AvatarController : Controller
{
    public ActionResult Index(string id, int s, string d, string r)
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۱۶ ۱۴:۲۲

تاثیر 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.

نویسنده: امیرحسین جلوداری
تاریخ: ۱۳۹۳/۱۰/۲۱ ۲۲:۲۷

با سلام؛ من وقتی روی اکشن متود Authorize میزارم و بعد میخواد فایل رو از طریق نرم افزاری مثه IDM دانلود کنم میبینم که اونو به صفحه‌ی login منتقل میکنه در حالی که میخوام دیالوگ login در نرم افزار IDM بیاد بالا!
چجوری میشه این کارو کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۲۱ ۲۲:۴۹

- دیالوگ لاگین در IDM برای حالت Basic authentication ظاهر می‌شود.
+ نرم افزار IDM اگر از طریق افزونه‌های آن (^) اقدام به دریافت این فایل کند، مشکلی نخواهد داشت؛ چون این افزونه‌ها اطلاعات سشن جاری کاربر را به برنامه، جهت شبیه سازی و استفاده‌ی مجدد منتقل می‌کنند (اطلاعاتی مانند کوکی‌ها و تمام مشخصات جاری صفحه‌ی لاگین کرده). به عبارتی اگر شخصی به سایت لاگین کند و از طریق مرورگری که افزونه‌ی IDM بر روی آن نصب است، اقدام به دریافت فایل کند (بر روی لینک کلیک کند تا مرورگر توسط افزونه‌ی مربوطه، درخواست را به برنامه ارسال کند)، دریافت فایل از دید او معمولی و مانند قبل خواهد بود.

عنوان: ModelBinder سفارشی در ASP.NET MVC
نویسنده: ناصر طاهری
تاریخ: ۱۳۹۲/۰۴/۲۰ ۸:۲۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

زمانی که درخواستی به سمت یک Action پارامتر دار ارسال میشود، قسمت ActionInvoker قبل از فراخوانی اکشن مربوطه، به دنبال Model Binder مناسبی برای داده‌های پارامترها می‌گردد و در صورت یافت نشدن، از ModelBinder پیش فرض ASP.NET MVC استفاده می‌کند.

اما وظیفه‌ی ModelBinder چیست ؟

ModelBinder 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 :

همانطور که می‌بینید تایپ کردن تاریخ به این صورت (2009/1/1 12:00:00 AM) ، 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 ، ModelBinder 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 ارثبری و متد BindModel آن را پیاده سازی کنیم.
ساختار این اینترفیس به شکل زیر است :

```
public interface IModelBinder
{
    object BindModel(ControllerContext controllerContext,
        ModelBindingContext bindingContext);
}
```

متد 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 پردازشش 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(HttpRequestBase 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 متد 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 کتابخانه‌ی 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

ممنون
بله. در صورتی که ModelBinder رو در Application_start() ریجیستر 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 ریجیستر نکنیم و مستقیما 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 مدلم تنظیم کنم. یعنی پراپرتی از جنس byte[] داریم که میخواهم توسط FileModelBinder بایند شود. FileModelBinder 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 متد BindProperty را override کرده و سپس اگر خاصیتی، مثلا خاصیت X بود (بر اساس مثلا یک ویژگی خاص که به آن انتساب داده شده)، آنگاه از Model binder سفارشی 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: ۱۳۹۳/۱۰/۰۱ ۱۰:۴۷

سلام؛ با تشکر از مقاله شما.
میخواستم بپرسم override کردن BindModel یا BindProperty برای زمانییه 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: ۱۳۹۳/۱۰/۰۱ ۱۱:۰

- 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، دقیقا یک لیست از شیء Value را دریافت می‌کنند.
- توضیحات تکمیلی آن در اینجا « ASP.NET Wire Format for Model Binding to Arrays, Lists, Collections, Dictionaries »

عنوان: **ایجاد لینک با یک تصویر بوسیله Html Helper**
نویسنده: مهراد
تاریخ: ۱۳۹۲/۰۵/۰۸ ۸:۵
آدرس: www.dotnettips.info
گروه‌ها: Extension Method, ASP.Net MVC, MVC

با Html Helper ما 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.
همان طور که در کد زیر مشاهده 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 می‌کنیم:

```
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. 4 پارامتر بعدی، پارامترهای ضروری هستند. یکی controller، پس از آن Action 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"})
```

در شکل بالا خروجی HTML تولید شده را مشاهده می‌نمایید.

نظرات خوانندگان

نویسنده: Hamid NCH
تاریخ: ۱۳۹۲/۰۵/۲۵ ۱۰:۱

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: ۱۳۹۲/۰۵/۲۵ ۱۰:۶

با عرض معذرت.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: ۱۳۹۲/۱۰/۰۷ ۱۴:۱۸

سلام
مرسی از مطالب مفیدتون. میشه راهنمایی کنید که چطوری توی یه پروژه MVC از این کد باید استفاده کرد؟

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۰۷ ۱۹:۳۶

یک کلاس استاتیک جدید درست کنید. بعد متد الحاقی ActionImage رو بهش اضافه کنید. حالا در کدهای Razor 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: ۱۳۹۲/۱۰/۰۸ ۹:۲۹

```
Compiler Error Message: CS0103: The name 'workout' does not exist in the current context
@using ActionImage.Extentions;
@Html.ActionImage("Workout", "Delete", new { id = @workout.Id }, "/images/bluetrash.png", "Delete", new
{ id="deletebutton"})
```

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۰۸ ۹:۵۱

بجای اون id هرچی دوست داشتی بنویس. احتمالا یک حلقه بوده که workout یکی از اشیاء اون هست. مهم نیست. جزو کار نیست.

نویسنده: رشوند
تاریخ: ۱۳۹۳/۰۴/۲۹ ۱۸:۴۷

سپاس, بسیار مفید بود.
البته در کد آقا مهرداد هم به صورت H نوشته شده که لطفا اصلاح بفرمایید...

نویسنده: رشوند
تاریخ: ۱۳۹۳/۰۵/۱۱ ۱۶:۳۰

یه سوال برام پیش اومد
فرض کنیم کاربری لاگین کرده رولش هم مثلا خبرنگاره,
در پروفایلش لیست خبر هایی که ثبت کرده را می‌بینه با توجه به مثالی که شما هم زدید , اگر مثلا عدد "1" لینک ویرایش (یا حذف) که نشانگر id است رو به صورت دستی(تغییر کد html و ...) تغییر بده, می‌تونه 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)؟
لطفا 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 رمزنگاری خودکار فیلدهای مخفی وجود دارد.
- بررسی‌های سمت سرور هم جای خودشان را دارند و لازم هستند.

عنوان: **بارگزاری PartialView با استفاده از jQuery در زمان اجرا**
نویسنده: پویا امینی
تاریخ: ۱۳۹۲/۰۵/۲۲ ۱۸:۲۰
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net MVC, jQuery, jQuery Ajax, MVC

مزیت استفاده از PartialViewها، ماژولار 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 پیاده 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 درIndex، این View را بارگزاری کنیم. برای این کار از متد 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 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);
});
    });
```

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:

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۵/۲۲ ۱۹:۵۹

مطالب تکمیلی که از همین روش استفاده می‌کنند:

به روز رسانی غیرهمزمان قسمتی از صفحه به کمک jQuery در ASP.NET MVC

پیاده 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 « بارگذاری »؟ البته 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 داره 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 اصلا 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 الحاق کنید تا به صورت خودکار در جزئی از صفحه آن، که بعدا به روز خواهد شد، نیز اعمال شود.
- راه دوم: اصلا از RenderSection در یک partial view که قرار است Ajax ایی بارگذاری شود، استفاده نکنید. معمولی این‌ها را الصاق یا تعریف کنید. مثل تعاریف یک HTML ساده. یک نفر هم اینجا براش HtmlHelper نوشته ولی نکته اصلی یکی است؛ الصاق و تعریف معمولی فایل‌های مورد نیاز.
- همچنین خود jQuery امکان بارگذاری اسکریپت‌ها را به صورت پویا دارد . زمانیکه complete عملیات Ajax ایی رخداد، متد getScript 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؟

ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۶/۱۳ ۹:۰

نحوه صحیح ارسال پارامترها توسط jQuery Ajax
ارسال کوئری استرینگ‌ها به همراه عملیات Ajax در jQuery

ضمنا بحث وب گرید ، خارج از موضوع مطلب جاری است. از پارامتر ajaxUpdateContainerId آن افزونه خاص باید برای مدیریت مسایل paging و sorting استفاده کنید.

نویسنده: مجتبی شایان فر
تاریخ: ۱۳۹۲/۰۶/۲۸ ۱۳:۴

باسلام 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 .
ممنون.

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۶/۲۸ ۲۰:۴

نحوه استفاده از افزونه Firebug برای دیباگ برنامه‌های ASP.NET مبتنی بر jQuery

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: ۱۳۹۲/۰۸/۱۶ ۱۰:۵۴

- برنامه را با فایرباگ دیباگ کنید تا مسیرهای درخواستی و خطاها را بهتر مشاهده کنید.
- نباید آدرس‌ها را مستقیما 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: ۱۳۹۲/۰۸/۱۷ ۱۱:۳۶

با 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 ان 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 میکنم چطور میتونم ان را حل کنم ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۸/۱۷ ۱۱:۴۸

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: ۱۳۹۲/۰۸/۲۱ ۱۱:۱۹

با عرض سلام و خسته نباشید
این روش فرقش با jquery.load('url') چیه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۸/۲۱ ۱۴:۲۴

load در پشت صحنه از همین متد ajax استفاده می‌کند. تمام متدهای کمکی از این دست فقط محصور کننده تابع ajax اصلی جی‌کوئری هستند.

نویسنده: رشنو
تاریخ: ۱۳۹۲/۱۲/۱۵ ۱۰:۳۶

با سلام و تشکر
لیست محصولات رو چطور باید پاس داد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۱۵ ۱۰:۵۸

به روش‌های متداول « بررسی نحوه انتقال اطلاعات از یک کنترلر به Viewهای مرتبط با آن »

عنوان: **یافتن اکشن متدهای به اشتباه کش شده در ASP.NET MVC**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۶/۱۱ ۲۳:۴۵
آدرس: www.dotnettips.info
برچسب‌ها: MVC, Security

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 محتوای حاصل از آن، در دفعه‌ی بعد فراخوانی این آدرس خاص، اصلا چرخه کاری یک کنترلر روی نداده و تمام مسایل توسط 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؟ مثلا 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 روی دهند. اما در بار دوم فراخوانی آدرس دریافت صفحه یا فایل، اصلا کار به فراخوانی کنترلر نمی‌رسد. به عبارتی کلیه کاربران سایت (اعم از لاگین شده، نشده، دارای دسترسی مشاهده صفحه یا آدرس امن و یا بدون دسترسی)، به این محتوای خاص بدون مشکلی دسترسی خواهند داشت (فقط کافی است که از آدرس نهایی به نحوی مطلع شوند).

سؤال: چگونه می‌توان کلیه اکشن متدهای 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)
        {
            // یافتن 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)
            {
                // یافتن کلیه اکشن متدهای کنترلر جاری
                var actionMethods = controller.GetMethods(BindingFlags.Public | BindingFlags.Instance |
BindingFlags.DeclaredOnly)
                                                .Where(method =>
typeof(ActionResult).IsAssignableFrom(method.ReturnType))
                                                .ToList();

                foreach (var method in actionMethods)
                {
                    // یافتن متدهایی 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 ای‌جکسی

                    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 اکشن متدها مدنظر هستند). اگر این اکشن متد یافت شده دارای ویژگی 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 متد امن، ارجاعی به اکشن متدی کش شده در کنترلری دیگر وجود داشت، این مورد هم یک باگ امنیتی است.
- کلیه کنترلرهای عمومی که دارای اکشن متدی امن هستند نیز باید جهت یافتن OutputCache بررسی شوند.

عنوان: مروری مقدماتی بر ساخت برنامه‌های موبایل در MVC4
نویسنده: وحید م
تاریخ: ۱۳۹۲/۰۶/۱۵ ۲۳:۱۵
آدرس: www.dotnettips.info
برچسب‌ها: MVC, Mobile Websites, Responsive Web Design

برای شروع، پروژه‌ای از نوع InternetApplication را انتخاب نموده مطابق شکل زیر:

سپس اگر صفحه _Layout.cshtml باز کنید، با متا تگی 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:

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 بواسط آنها می‌توان مثلا UserAgent ای را که می‌خواهیم، مشخص کنیم و سپس در DisplayModelProvider آن را بعنوان Instance ای درج و سپس تابع InitialDisplayModeProvider را در Application_start تنظیم کنیم.

حال می‌بایست Pageهای مربوط به هر کدام را طراحی کرده و با پسوند مثلا (Phone-Tablet)ذخیره کنیم.

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 وبا jquerymobile روی آن کار کنید.

عنوان: **اجرای برنامه‌های ASP.NET توسط Mono در Ubuntu**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۷/۰۱ ۱۰:۰
آدرس: www.dotnettips.info
برچسب‌ها: ASP.Net, MVC, Mono, Linux

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 .

پیشنیازها

دو پروژه خالی ASP.NET Web forms و ASP.NET MVC 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؛ خصوصا نگارش MVC که به همراه یک سری کتابخانه جانبی است).

برای انتقال فایل‌ها به لینوکس، اگر از VMWare workstation برای اجرا و آزمایش 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 ASP.NET Web forms 4 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 حتما 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.

پیشنیازهای اینکار مانند Git و Mono، در قسمت قبل دریافت شدند. سپس فرامین 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 خواهید 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 آن‌را در monodevelop باز کرده و پروژه را build کنید). در این حالت دو فایل Mono.WebServer.dll و Mono.WebServer.XSP.exe در پوشه xsp/src/Mono.WebServer.XSP/bin/Debug ظاهر 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 پوشه برنامه MVC خود کپی کنید و سپس دستور ذیل را اجرا نمائید:

```
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 خواهید 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 پوشه 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 اسمبلی‌های 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 کفایت می‌کنه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۷/۰۱ ۱۷:۵۶

- MySQL هم خوبه (البته مجوز استفاده‌اش رو هم باید مدنظر داشته باشید). هرچند 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 oracle و Ms SqlServer 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 نحوه‌ی 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 صرفا 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
```

mod_mono درون پروسه آپاچی اجرا شده و تمام درخواست‌های رسیده به یک وب سایت ASP.NET را به mod-mono-server که کار 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 ASP.NET 4 تنظیم شده‌است.

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 پوشه‌های 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 ببندید. 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 آن‌را 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 File browser 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/www کپی کرد.
پس از کپی دو برنامه Web forms و MVC ابتدای بحث، نیاز است مجددا 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 /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/webforms_test و برنامه MVC در مسیر http://127.0.0.1/mvc_test قابل دسترسی است.

خلاصه بحث

پس از نصب وب سرور آپاچی و ماژول 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>
```

اجرای ASP.NET MVC بر روی لینوکس

نظرات خوانندگان

نویسنده: مهدی پایروند
تاریخ: ۱۳۹۲/۰۷/۰۳ ۹:۴۷

در مورد هاست کردن سرویس‌های WCF هم اگه ممکنه توضیح بدین.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۷/۰۴ ۱۴:۳

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 ایی قابل استفاده خواهد بود (فرض بر این بود که یک Ajax Enabled WCF Service را در VS.NET ایجاد و به لینوکس منتقل کردید).

نویسنده: ایمان اسلامی
تاریخ: ۱۳۹۲/۱۲/۲۰ ۸:۲۷

میخواستم بدونم در شرایط مساوی نصب یک برنامه asp.net روی iis از نظر performance بهتره یا روی apache؟

عنوان: **فعال‌سازی استفاده از Session در ASP.NET MVC 4 API Controller ها**
نویسنده: بهمن خلفی
تاریخ: ۱۳۹۲/۰۷/۱۷ ۲۱:۳۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, Session, ASP.Net MVC 4, routing, Web API

در پروژه فروشگاهی تحت Asp.Net MVC 4 بهترین روش برای ایجاد یک سبد خرید استفاده از یک Api Controller می‌باشد.

من در پروژه‌ای که در MVC 3 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 برنامه‌ای که ساختیم کار کند باید در Global.asax این کدها را درج کنیم (دوباره تاکید می‌کنم این نکته فقط در MVC 3 کار می‌کند)

```
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 برنامه‌های Asp.NET MVC 3 امکان استفاده از Session در API Controllerها در دسترس قرار میگیرد.
اما این کار در ASP.NET MVC 4 قابل استفاده نیست و خیلی بهتر و راحتتر انجام می‌شود (میتوانید به WebApiConfig.cs در فولدر 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 Controllerهای MVC4، 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.SetSessionStateBehavior(
        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

لطفا 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 پروسه ای زمان بر بوده و کاربر پسند نیست در حالی که در فروشگاههای متن باز مشابه این عمل بصورت زیبا و کاملا 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 Actionهای 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 نمائید 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 Controllerها 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. خصوصا اینکه یک شخص آنچنان اقلام زیادی را که هربار خرید نمی‌کند.
همچنین HTML5 local storage نیز مفید است.

نویسنده: بهمن خلفی
تاریخ: ۱۳۹۲/۰۷/۲۰ ۸:۵۷

بله دوست عزیز ، استفاده از Session مزایا 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)

نویسنده: 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

عنوان: **ساخت منوهای چند سطحی در ASP.NET MVC**
نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۲/۰۷/۱۹ ۱۱:۲۵
آدرس: www.dotnettips.info
گروه‌ها: Entity framework, MVC, Twitter Bootstrap

پیش نیاز مطلب جاری مطالب زیر می‌باشند:

1- EF Code First #8
2- مباحث تکمیلی مدل‌های خود ارجاع دهنده در EF Code First
3- نگاهی به اجزای تعاملی Twitter Bootstrap

هدف از مطلب جاری نحوه نمایش منوی‌های چند سطحی می‌باشد، 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 می‌کنید، مدل ما شامل مشخصات گروه محصولات می‌باشد که به صورت خود ارجاع دهنده (خاصیت Parent به همین کلاس اشاره میکند) تعریف شده است. در مورد خواص مدل‌های خود ارجاع دهنده، مطالبی را در سایت مطالعه کردید (خواص مربوط در مطالب گفته شده دقیقاً به همان صورت می‌باشد و نیازی به توضیح اضافه‌تری نیست).
هدف از این بحث، نحوه نمایش گروه محصولات داخل منو به صورت چند سطحی می‌باشد، جهت نمایش می‌بایست از تکنیک recursive function استفاده کنید، ابتدا در نظر داشته باشید که ساختار منوی تشکیل شده می‌بایست 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: ۱۳۹۲/۰۷/۲۲ ۱۱:۱۱

با سلام

من این برنامه رو اجرا کردم ولی روی خط 5 پیغام خطا System.NullReferenceException: Object reference not set to an instance of an object.

رو میگیرم .

اگر امکانش هست میتونید پروژه این مثال رو بگذارید.

ممنون

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۲/۰۷/۲۲ ۱۱:۵۶

شما Query رو به View مربوطه پاس دادید؟

من توی لایه Service متد GetAll رو به این صورت تعریف کردم :

```
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: ۱۳۹۲/۰۷/۳۰ ۱۲:۱۱

سلام

اگر امکان دارد مثالی از درج ، ویرایش یا حذف زیر مجموعه‌ها را بیان کنید یا اگر منبع مفیدی می‌شناسید برای ویرایش و کار با navigation peroperty‌ها لطفا معرفی کنید

با سپاس فراوان

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۲/۰۷/۳۰ ۱۴:۲

سلام،

به عنوان مثال من برای ثبت یک Category جدید از TreeView استفاده میکنم به اضافه یک مورد که تعیین کننده Child یا Parent بودن Category است.

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۰۹/۰۳ ۲۳:۵۴

جناب مهندس ممنون از زحمت شما.
یک سوال؟
در بوت استرپ منوی چند سطحی نمی‌توان درست کرد ایا کد خاصی اضافه کردین یا از پلاگین خاصی استفاده میکنین؟

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۲/۰۹/۰۴ ۰:۴

من در این مثال از ورژن 2 بوت استرپ استفاده کرده ام که دارای کلاس dropdown-submenu می‌باشد. این کلاس ظاهراً 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 ۱۴:۲۵

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 هربار لود صفحه درخواست به بنک ارسال نشه

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۳/۰۲ ۱۴:۴۰

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 Interceptors برای حذف کدهای تکراری کش کردن اطلاعات در لایه سرویس برنامه
و ...

نویسنده: رضایی
تاریخ: ۱۳۹۳/۰۳/۰۳ ۰:۵۴

ممنون؛ از حالت AOP Interceptors استفاده کردم خیلی جالب بود.
فقط یه راهنمایی میخواستم به نظر شما برای کش کردن منوهای پویا چه زمانی میتونه مناسب باشه؟

```
[CacheMethod(SecondsToCache = ?)]
```

باز هم تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۳/۰۳ ۱:۲

در سایت جاری اطلاعات آماری پویای منوی سمت راست صفحه، 15 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 متد 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 متد 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 رو غیرفعال کرد و پس از اولین لود دوباره اطلاعات کش شوند

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۲۸ ۱۳:۳۳

- « CacheManager.InvalidateCache »
- « EfHttpRuntimeCacheProvider.InvalidateSecondLevelCache »

نویسنده: حسین صفدری
تاریخ: ۱۳۹۳/۱۰/۲۳ ۱۱:۴۸

با سلام جسارتا جهت RTL منو بوت استرپ 3 به حالت چند سطحی باید کد بالا را به حالت زیر تغییر داد

```
.dropdown-submenu{position:relative}
.dropdown-submenu > .dropdown-menu{top:0;right:100%;margin-top:-6px;margin-right:-1px;-webkit-border-
radius:0 6px 6px 6px;-moz-border-radius:0 6px 6px;border-radius:0 6px 6px 6px}
.dropdown-submenu:hover > .dropdown-menu{display:block}
.dropdown-submenu > a:after{display:block;content:" ";float:left;width:0;height:0;border-
color:transparent;border-style:solid;border-width:5px 5px 5px 0;border-right-color:#ccc;margin-
top:5px;margin-left:-10px}
.dropdown-submenu:hover > a:after{border-left-color:#fff}
```

```
.dropdown-submenu.pull-Right{float:none}
.dropdown-submenu.pull-Right > .dropdown-menu{right:-100%;margin-right:10px;-webkit-border-radius:6px 0
6px 6px;-moz-border-radius:6px 0 6px 6px;border-radius:6px 0 6px 6px}
```

عنوان: **طراحی ValidationAttribute دلخواه و هماهنگ سازی آن با ASP.NET MVC**
نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۰۷/۲۱ ۱۵:۳۵
آدرس: www.dotnettips.info
گروه‌ها: Entity framework, MVC, ASP.Net MVC, jQuery

در سری پست‌های آقای مهندس یوسف نژاد با عنوان Globalization در ASP.NET MVC روشی برای پیاده سازی کار با Resourceها 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 Resourceها از طریق Attributeها 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 پیاده سازی 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 MVC و Entity Framework 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 برنامه در زمان ثبت داده‌ها هیچ گونه خطایی 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 خطاها را نمایش خواهد داد.

تا اینجای کار روال عادی همیشگی است. اما برای طراحی Attribute 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 Resourceهای 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. Validator 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 خطا 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 Unobtrusive jQuery پیام خطای مورد نظر به کنترل ورودی مورد نظر (مانند تکست باکس) اعمال می‌شود. مثلا در این مثال خصوصیت data-val-required به input هایی که قبلا در مدل ما Reqired 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 ولیدتور ابتدا کلید مورد نظر با توجه به زبان فعلی از دیتابیس بازیابی شده و در متادیتابی مدل ما قرار می‌گیرد. به جای استفاده از Resourceها می‌توان پیام‌های خطای دلخواه را در دیتابیس ذخیره کرد و در مواقع 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 ^ و ^

نظرات خوانندگان

نویسنده: امیر خلیلی
تاریخ: ۱۳۹۲/۰۸/۰۴ ۸:۵۹

یک مشکل اساسی وجود داره
هنگامی که با استفاده از جاواسکریپت اعتبارسنجی انجام میشه کاربر یا اون شخصی که قراره کدهای مخرب را وارد کنه میتونه استفاده از فایل‌های 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.

عنوان: **بهینه سازی برنامه‌های وب ASP.NET برای موتورهای جستجو (SEO)**
نویسنده: محسن نجف زاده
تاریخ: ۱۳۹۲/۰۷/۲۴ ۱۴:۵۰
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, C#, MVC, SEO

می‌دانیم بهینه‌سازی موتورهای جستجو (به انگلیسی: Search engine optimization (SEO)) که گاهی در فارسی به آن سئو نیز گفته می‌شود، عملیاتی است برای بهبود دید یک وب‌گاه یا یک صفحهٔ وب، در صفحه نتایج موتورهای جستجو که می‌تواند طبیعی و یا الگوریتمی باشد. این عملیات برای وبمسترها یکی از عوامل مهم و حیاتی بدست آوردن کاربران جدید از موتورهای جستجو است.

اگر چک لیست‌های SEO وب سایت ها را مشاهده کنیم، می‌توانیم آن‌ها را در دو دسته‌ی کلی بهینه سازی درونی و برونی وب سایت در نظر بگیریم :
Off-Page Optimization یا برونی ، که بیشتر بر دوش مشاوران سئو و خود مدیران وب سایت است . (link building ، فعالیت در شبکه اجتماعی و ...)
و اما در حوزه **On-Page Optimization** یا درونی که بخش‌های مهمی از آن وظیفه‌ی مابرنامه نویس‌ها است . (H1 Tag ، URL Naming ، Meta Tags ، عنوان صفحه و ...)
[البته عامل درونی بهینه سازی محتوا (Content Optimization) که مهمترین عامل در الگوریتم‌های نسل جدید موتورهای جستجو و همچنین الگوریتم جدید گوگل (و +) به حساب می‌آید بر عهده مشاوران سئو و خود مدیران وب سایت می‌باشد]

در ادامه به ارائه چند راهکار جهت بهینه سازی برنامه‌های وب ASP.NET مان برای موتورهای جستجو می‌پردازیم:

1.متدی برای ایجاد عنوان سایت

```
private const string SeparatorTitle = " - ";
    private const int MaxLenghtTitle = 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 <= MaxLenghtTitle ? title.Length :
MaxLenghtTitle).Trim();

        return title;
    }
```

نکته :
MaxLenghtTitle پیشنهادی برای عنوان سایت 60 می‌باشد.

2.متدی 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 MaxLenghtTitle = 60;
private const int MaxLenghtDescription = 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 <= MaxLenghtTitle ? title.Length :
MaxLenghtTitle).Trim();
        description = description.Substring(0, description.Length <= MaxLenghtDescription ?
description.Length : MaxLenghtDescription).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 :

MaxLenghtDescription پیشنهادی برای متاتگ توضیح سایت 170 می باشد.

آشنایی با متاتگ‌ها (Meta tags) و کاربرد آنها در صفحات وب (HTML)

برای کاربرد allowIndexPage و allowFollowLinks هم می‌توانید به لینک بالا و بررسی متاتگ robots بپردازید.

با توجه به اهمیت شبکه‌های اجتماعی متاتگ‌های شبکه‌های اجتماعی (+ و +) را هم نباید از قلم انداخت.

برای دریافت Description نوع سفارشی CacheControlType از پروژه متدهای الحاقی علیرضا اسم رام استفاده کردم.

3.متدی برای ایجاد Slug (اسلاگ آدرسی با مفهوم برای بکار بردن در URL ها است که دوست‌دار موتورهای جستجو می‌باشد)

```
private const int MaxLenghtSlug = 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 <= MaxLenghtSlug ? slug.Length : MaxLenghtSlug).Trim();

        return slug;
}

private static string RemoveAccent(string text)
{
        var bytes = Encoding.GetEncoding("UTF-8").GetBytes(text);
        return Encoding.UTF8.GetString(bytes);
}
```

نکته :

MaxLenghtSlug پیشنهادی برای عنوان سایت 45 می‌باشد.

نمونه ای از کاربرد توابع :

```
Head.InnerHtml = SEO.GenerateMetaTag
                            (
                                title: SEO.GeneratePageTitle(".NET Tips", "آرشیو مطالب", "ASP.NET MVC
#1"),
                                description: "چرا ASP.NET MVC با وجود فریم ورک پخته‌ای به نام ASP.NET web
forms، مطرح می‌شود این است: «برای چی؟». بنابراین تا به این سؤال ASP.NET MVC 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
                            );
```

و خروجی در Page Source :

```
<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="چرا ASP.NET MVC با وجود فریم ورک پخته‌ای به نام ASP.NET web forms،
مطرح می‌شود این است: &#171;برای چی؟&#187;. بن ASP.NET MVC 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 خواهند 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 می‌کنم :

```
<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 یا <label> احاطه می‌شود. (+)

نویسنده: پوریا منفرد
تاریخ: ۱۳۹۲/۱۰/۱۴ ۱۲:۱۵

با سلام
امکان داره این خط کلاسش رو هم بزارید بنده دقیقا نتونستم با متد الحاقی 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 می‌کنم در ASP.net Webform و فرمی رو بهش Connect کردم که:
1- در MPage یک سری متا تگ‌ها هست + JavaScript + CSS
2- وقتی در یک webForm متصل به Mpage کد:

```
Head.InnerHtml = SEO.GenerateMetaTag ....
```

رو قرار می‌دهم تمام تگ‌های Head حتی MasterPage هم پاک می‌شه
اینم می‌دونم این کلاس این کارو می‌کنه
راه حل چیه که تگ‌ها پاک نشه ؟ مخصوصا آدرس‌های CSS و JavaScript ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۰/۱۴ ۱۳:۵۱

کمی بالاتر توضیح دادند ؛ استفاده از یک Literal برای مثال.

نویسنده: مهرداد پاک دل
تاریخ: ۱۳۹۲/۱۱/۰۲ ۱۶:۱۶

من این توابع را خواندم که در مورد seo نوشته بودید ولی متاسفانه نحوه استفاده از آن هارا نفهمیدم یعنی کجا تعریف میشن و از کجا پارامترهای خود را دریافت می‌کنند. لطف کنید اگر امکان داره یه پروژه کوچک همه این‌ها استفاده شده باشه اگر هست بگذارید .
نکته دیگه شما عنوان هر پست را داخل urlها نمایش میدهید منظور اینکه جزئی از url سایت است این کار را چگونه انجام می‌دهید

```
mysite.info/post/1
```

```
dotnettips.info/post/1200/seo
```

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۱/۰۲ ۱۶:۳۸

- پروژه Iris هست. به section metatags اون دقت کن در فایل‌های View.

- مثلا متد Head.InnerHtml = SEO.GenerateMetaTag عنوان شده در این مطلب باید در Page_Load یک وب فرم فراخوانی شود. Id مطلب رو دارید. عنوان و متن و سایر مشخصات اون رو از دیتابیس دریافت کنید و بعد فقط یک جایگذاری است در متد تهیه شده.

- به اینکار Routing و Url rewrite می‌گن. بحثش در MVC و وب فرم‌ها کمی با هم فرق می‌کنه.

نویسنده: امین قادری
تاریخ: ۱۳۹۳/۰۵/۰۹ ۳:۳۵

لطفا مثالی را در مورد نحوه‌ی استفاده از " **متدی برای ایجاد Slug** " در عمل ارسال نمائید. با تشکر

نویسنده: موسوی
تاریخ: ۱۳۹۳/۰۹/۲۲ ۱۸:۹

سلام منم کدها رو تویه فایل csبه 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 می‌ده » برای رفع مشکل کافی نیست. چه خطایی می‌ده 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 headهم از یه لیترال استفاده کردم وخیلی خوب جواب داد.زنده باشین

نویسنده: محمدرضا امینی
تاریخ: ۱۳۹۳/۱۲/۰۳ ۱۱:۵۹

سلام.
هنگام جستجو در گوگل گاهی اوقات سایتی باز میشه که خود اون سایت هم کلمه کلیدی شما رو جستجو کرده و نتایج رو به شما نشون میده.
مثلا یک فروشگاه خوب هنگامی که در گوگل قیمت رم لپ تاپ رو جستجو کنید چنین صفحه ای داره.
به نظر شما دوستان چطور میشه چنین صفحه ای در سایت خود داشته باشیم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۲/۰۳ ۱۵:۷

- ابتدا یک پروفایل Google analytics ایجاد کنید و سپس اسکریپت آن‌را به سایت خودتان اضافه کنید.
- چند روز بعد که به آمار آن مراجعه کنید، می‌توانید لیست جستجوهای گوگل منتهی به سایت خودتان را به همراه واژه‌های کلیدی مرتبط، گزارش گیری کنید.

- بر اساس این واژه‌های کلیدی، برای محصولات خودتان برچسب درست کنید یا آن‌ها را گروه بندی کنید. گوگل بر این اساس در دفعات آتی، نتایج جستجوی دقیق‌تری را به کاربران ارائه می‌دهد.

نویسنده: محسن نجف زاده
تاریخ: ۱۳۹۳/۱۲/۰۳ ۱۵:۲۵

پیرو تکمیل صحبت آقای نصیری،
احتمالا جستجو در وب سایت مورد نظر در دو صفحه مجزا طراحی شده :
1.جستجوی‌های طبقه بندی شده بر اساس اطلاعات استخراجی شان از ابزارها ارزیابی وب سایت‌ها مانند Google analytics (راهنما در کامنت قبل)
2. جستجوی لحظه ای در محتوای سایت
این نکته در متا تگ‌های ایجاد شده نیز قابل مشاهده است :

www. .com/Search/Category-Laptop-RAM/#/Category-Laptop-RAM/

```
<title> جستجو در گروه رم لپ تاپ </title>
<meta content="جستجو در گروه رم لپ تاپ" name="description">
<meta content="جستجو در گروه رم لپ تاپ" name="keywords">
```

www. .com/Search/#/Keyword-رم لپ تاپ/

```
<title> جستجو </title>
```

تعداد نتایج یافت شده متفاوت این دو صفحه (با یک عنوان جستجو) نشان از تایید مطلب دارد.

نویسنده: محمدرضا امینی
تاریخ: ۱۳۹۳/۱۲/۱۹ ۱۶:۵۱

بر فرض مثال ما می‌خواهیم زمانی که از گوگل شخصی به سایت ما هدایت می‌شود با کلمه کلیدی ای که وارد سایت شده به طور خودکار در سایت جستجو کنیم.
کلا چطور میشه متوجه شد که کاربری از گوگل به سایت ما وارد شده و با چه جستجویی؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۲/۱۹ ۱۹:۱۴

Request.UrlReferrer را آنالیز کنید. مقدار آن مشخص می‌کند که مثلا شخص از گوگل است یا سایت‌های جستجوی دیگر. بعد بر این اساس، کوئری استرینگ q آن‌را یافته و سپس شخص را به صفحه‌ی جستجو هدایت کنید:

```
// for Google
var urlReferrer = Request.UrlReferrer.ToString();
var query = HttpUtility.ParseQueryString(urlReferrer);
var searchQuery = query["q"];
```

یک نمونه‌ی کاملتر

نویسنده: علی یگانه مقدم
تاریخ: ۱۳۹۳/۱۲/۲۰ ۲۰:۲۹

یک سوال داشتم
اینکه این گونه سایت‌ها از این روش برای افزایش بازدید استفاده می‌کنند یک نوع کلاه برداری در SEO به شمار نمیاد؟
چون که عموما این سایت‌ها هیچ محتوای مرتبطی ندارن و به خاطر افزایش بازدید اینکارو می‌کنن و باعث به هم ریختن نتایج جست و جو میشن تا جایی که گاهی اوقات می‌بینی چند صفحه اول تمام لینک‌ها از این قبیل سایت هاست و گوگل هم بارها هشدار داده سایت هایی که کلاه برداری می‌کنند و محتوا ندارن رو جریمه می‌کنه ولی به تا به حال یک بار هم این حرف عملی نشده و همچنان همان سایت‌ها هم به فعالیت ادامه میدن و باعث به هم ریختگی میشن

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۱۲/۲۱ ۹:۵۶

هرکسی از ظن خود شد یار من ...

عنوان: CheckBoxList برای فیلد Enum Flags مدل در ASP.Net MVC
نویسنده: پژمان پارسائی
تاریخ: ۱۳۹۲/۰۷/۳۰ ۲۳:۰
آدرس: www.dotnettips.info
گروه‌ها: C#, ASP.Net MVC, Enum, ASP.Net MVC 4, MVC

قبلا مطالبی در سایت راجع به نوع داده شمارشی یا Enum و همچنین CheckBoxList و RadioButtonList 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 بدهید تا بتواند چندین گزینه را برای یک فیلد انتخاب کند. به عنوان یک مثال ساده فرض کنید گزینه ای 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 مثلا می‌توانیم به صورت زیر تقسیم بندی کنیم :

پنبه (Cotton)
ابریشم (Silk)
پشم (Wool)
ابریشم مصنوعی (Rayon)
پارچه‌های دیگر (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. مثلا 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 View Model 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 View Model 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. مثلا متغیر cotWool هم دارای مقدار Fabric.Cotton و هم دارای مقدار Fabric.Wool هست. مقدار عددی معادل متغیر cotWool برابر 5 هست که از جمع مقدار عددی Fabric.Cotton و Fabric.Wool به دست آمده است.

حال فرض کنید فیلد Fabric از View Model ذکر شده (کلاس MyViewModel) را به صورت لیستی از چک باکس‌ها نمایش دهیم. مثل زیر:

شکل (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 گزینه هایی 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 یه Enum به یک CheckBoxList و همچنین شامل Model Binder 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 html helper 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.ToDescription());
                }
            }
            sb.AppendLine(label.ToString());

            // Add line break.
            sb.AppendLine("<br />");
```

```
        }
    }

    return new HtmlString(sb.ToString());
}
```

در کدهای بالا از متد الحاقی ToDescription نیز برای 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(Descripti
onAttribute), false);
    return attributes.Length > 0 ? attributes[0].Description : value.ToString();
}
```

برای استفاده از این Html Helper در View کد زیر را می‌نویسیم:

```
@Html.CheckBoxesForEnumFlagsFor(x => x.Fabric)
```

که باعث تولید خروجی که در تصویر (الف) نشان داده شد می‌شود. و 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 متد Application_Start فایل Global.asax فراخوانی کنیم:

```
ModelBinders.Binders.Add(typeof(Fabric), new FlagEnumerationModelBinder());
```

مشاهده می‌کنید که در اینجا دقیقا مشخص کرده ایم 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 پراپرتی آن برگشت دهیم. مثلا بدین صورت:

```
public class ModelEnums
{
    public static IEnumerable<Type> Types
    {
        get
        {
            var types = new List<Type> { typeof(Fabric) };
            return types;
        }
    }
}
```

سپس به متد 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:

شکل (ب)

و سپس فرم را پست کنید، موردی شبیه زیر مشاهده می‌کنید:

شکل (ج)

همچنین مقدار عددی معادل در این جا 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 شده اند.

پروژه مربوط به این مثال را از لینک زیر دریافت کنید:
MvcEnumFlagsProjectSample.zip

پی نوشت : پوشه‌های bin و obj و packages جهت کاهش حجم پروژه از آن حذف شده اند. برای بازسازی پوشه packages لطفا به مطلب بازسازی کامل پوشه 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 مثلا لیستی از یوزرها رو 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

سؤال شما بیشتر به مطلب « CheckBoxList در ASP.NET MVC » مرتبط است تا enum ایی که ویژگی flag دارد. ضمنا گرید تلریک مستندات خوبی دارد که بهتر است به آن مراجعه کنید (مثال Ajax CheckBoxes هست که کدهای View و کنترلر 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 اسمبلی‌های Resource استفاده کنیم چی؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۸/۰۲ ۱۲:۳۲

مراجعه کنید به پیشنیازهای این مباحث تکمیلی. مانند:
تهیه سایت‌های چند زبانه و بومی سازی نمایش اطلاعات در ASP.NET MVC
Globalization در ASP.NET MVC
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

عنوان: **چطور مسیریابی‌های ASP.NET MVC را دیباگ کنیم؟**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۸/۰۸ ۸:۵۵
آدرس: www.dotnettips.info
گروه‌ها: MVC

سؤال: من برای تهیه sitemap برنامه، یک route سفارشی نوشته‌ام 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 اصلا پاسخ نمی‌دهد. چرا؟ نحوه‌ی 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 بلافاصله بتوانید مشکل را حدس بزنید. اما اگر تعداد مسیریابی‌های سفارشی تعریف شده زیاد باشد، اینکار ساده نیست و حتما 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 فعال و برنده آن‌را مشاهده خواهید کرد. برای مثال در صفحه اول برنامه داریم:

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 برعهده 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:

localhost:1032/sitemap.xml

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

Key	Value
controller	sitemap.xml
action	Index
id	

Data Tokens

Key	Value

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 اصلا 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 مجددا مسیر sitemap.xml را درخواست کنیم، به تصویر ذیل خواهیم رسید:

localhost:1032/sitemap.xml

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: sitemap.xml

Route Data

Key	Value
controller	Sitemap
action	index
name	
area	

Data Tokens

Key	Value

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؟ مثلا اگر به مسیر index کنترلر home مراجعه کنیم، مشکلی نخواهد بود؟

پاسخ: خیر. برای آزمایش آن باز هم به افزونه RouteDebug مراجعه خواهیم کرد:

localhost:1032/#

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

Key	Value
controller	Home
action	Index
id	

Data Tokens

Key	Value

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: ۱۳۹۳/۰۱/۱۸ ۲۳:۵۲

یک نکته‌ی تکمیلی

افزونه‌ی Routing Assistant نیز برای این منظور مفید است.

عنوان: ذخیره TreeView ساخته شده توسط KendoUI در Asp.net MVC
نویسنده: حسین حسینی 128
تاریخ: ۱۳۹۲/۰۸/۱۳ ۱۲:۵
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;
        -mox-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 کردن 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 درخط اول تمام عناصر داخلی treeview را گرفته ، به صورت یک datasource قابل انقیاد درآورده وسپس datasource آن را به یک نمایش قابل تجزیه تبدیل میکند و به متد 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 درحال 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 رشته 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 رشته 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 چهارم45" 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 جداسازی 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;
                }
            }
        }
```

calculte_serialized با گرفتن 0 کار خود را شروع میکند یعنی از گره هایی که پدر ندارند و تمام idها را همراه با پدرشان در یک دیکشنری میریزد و هرکجا 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 می‌شود و بعد گزینه هایی که تیک دارد را در جدولی دیگر ذخیره می‌کنیم

عنوان: **تنظیمات امنیتی Glimpse**
نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۲/۰۸/۱۴ ۰:۴۵
آدرس: www.dotnettips.info
برچسب‌ها: ASP.Net, MVC, Glimpse, Profiler

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 به 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 رجیستر شدن ندارد و تشخیص آن توسط Glimpse به صورت خودکار انجام می‌شود.

عنوان: **سفارشی سازی Binding یک خصوصیت از طریق Attributes**
نویسنده: محمد رعیت پیشه
تاریخ: ۱۳۹۲/۰۸/۲۵ ۲۲:۰
آدرس: www.dotnettips.info
برچسب‌ها: MVC, Enum, custom model binder

اگر با MVC کار کرده باشید حتما با ModelBinding آن آشنا 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 بایند 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:

1) یک مدل بایندر جدید را با پیاده سازی IModelBinder تهیه کنیم. (در این حالت ما مجبوریم که مدل بایندر را از ابتدا جهت بایند کردن کلیه مقادیر شی مدل خود، بازنویسی کنیم و در واقع امکان انتساب آن‌را در سطح فقط یک خصوصیت نداریم.) (نحوه پیاده سازی قبلا در اینجا مطرح شده)

2) ModelBinder 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 DropDownFor معادل را قرار بدید که به طور خودکار رشته انتخاب شده را به یک خصوصیت مدل که از نوع بایت هست بایند 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 =
```

```
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 می‌کنیم. این کد فقط یک بار نوشته می‌شود و از این به بعد هر بایندر سفارشی که بسازیم به بایندر پیشفرض اضافه خواهد شد.

حالا از طریق کدهای زیر، ما بایندر سفارشیِ خصوصیتِ خودمان را به کلاس اضافه می‌کنیم :

```
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(Enums.AccountType),
request.Form["AccountType"]);
                propertyDescriptor.SetValue(bindingContext.Model, accountType);

                return true;
            }
            return false;
        }
    }
```

در کد بالا ما مقدار رشته‌ای را که از DropDownListFor ارسال شده، به مقدار عددی متناظر تعریف شده آن در Enum تبدیل می‌کنیم و آن را به خصوصیت مورد نظر بازگشت می‌دهیم، از این به بعد فقط برای فیلدی که به شکل زیر نشانه گذاری شده باشد، از این کلاس بایندر سفارشی استفاده می‌کنیم 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 به‌درستی کار می‌کند ... تا اینکه شما تصمیم میگیرید که از jquery.validate.unobtrusive برای اعتبار سنجی سمت کاربر استفاده کنید و می‌بینید به DropDownListFor شما هم ایراد میگیرید که حتما باید از نوع عددی باشد
The field نوع کاربر : must be a number. برای حل این مشکل هم باید به صورت دستی validation سمت کاربر رو برای این DropDownListFor غیرفعال کرد. برای این منظور باید کدهای DropDownListFor که در صفحه گذاشتید را به شکل زیر تغییر بدید:

```
@Html.DropDownListFor(model => model.AccountType, new
SelectList(Enum.GetNames(typeof(Enums.AccountType))),new Dictionary<string, object>() {{ "data-val",
"false" }})
```

نظرات خوانندگان

نویسنده: مهران بادامی
تاریخ: ۱۳۹۲/۰۸/۲۶ ۲۰:۲۴

سلام
من یهBinding نوشتم برای تاریخ که شمسی از کاربر گرفته به میلادی میدهد برای دخیره تو DB ولی وقتی میخواهم تاریخ که تو دیتابیسم میلادی هست را نشان بدهم همون میلادی نشون میده برای نمایش به شمسی چه کنم؟

نویسنده: محمد رعیت پیشه
تاریخ: ۱۳۹۲/۰۸/۲۸ ۱۰:۲۸

_ یک راه:
ایجاد یک HTML Helper سفارشی.
http://www.dotnettips.info/post/811/asp-net-mvc-8

عنوان: **Bundle کردن فایلهای LESS در MVC**
نویسنده: شهره مرتضوی
تاریخ: ۱۳۹۲/۰۹/۱۲ ۱۰:۵
آدرس: www.dotnettips.info
برچسب‌ها: CSS, ASP.Net MVC 4, LESS, Bundling, MVC

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... رفته و dotless 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- فولدری برای فایلهای .Less خود ایجاد کنید:

4- به App_Start/BundleConfig.cs رفته و کدهای ذیل در آن اضافه کنید:

```
var lessBundle = new Bundle("~/Content/Less").IncludeDirectory("~/Content/MyLess", "*.less");
            lessBundle.Transforms.Add(new LessTransform());
            lessBundle.Transforms.Add(new CssMinify());

            bundles.Add(lessBundle);
```

Bundling 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

عنوان: **TwitterBootstrapMVC**
نویسنده: رضا رضایی
تاریخ: ۱۳۹۲/۰۹/۱۲ ۲۳:۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, ASP.Net MVC 4, Mvc helper, Twitter Bootstrap, MVC 4, HTML helpers

TwitterBootstrapMVC یا به اختصار BMVC یک کتابخانه از Helper های مفید برای ساده سازی استفاده از Twitter Bootstrap در MVC می‌باشد .

در این کتابخانه امکانات مختلف Bootstrap از طریق Helper های نوشته شده برای MVC براحتی قابل استفاده می‌باشد و فرایند کد نویسی را ساده‌تر و در عین حال خواناتر می‌کند ، Helper های موجود در این کتابخانه به صورت زنجیره ای (fluent syntax) نوشته شده که استفاده از آن را سهولت می‌بخشد .

برای استفاده از آن در mvc 4 کافی است بعد از پیکر بندی Bootstrap (راهنمایی) به کتابخانه witterBootstrapMvc T رفرنسی ایجاد کنید و با استفاده از این راهنما نحوه استفاده را فرا گیرید . همچنین می‌توانید آن را از طریق NuGet بارگذاری نمایید .

نسخه‌ی MVC4 آن‌را در اینجا برای شما نیز آپلود نمودم در زیر نمونه ای 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!
ممنون.

نویسنده: افشار محبی
تاریخ: ۱۳۹۲/۰۹/۱۳ ۱۲:۵۷

تا حالا فرصت نشده که به طور اساسی با Twitter Bootstrap درگیر بشوم. اما وجود این تعداد راهنما 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 (مجانی نیست). البته فایل‌هایی که ایشون پیوست کردند به نظر محدودیت ندارند.
- ضمنا باز هم هستند یک سری Wrapper برای بوت استرپ که می‌توانند مورد استفاده قرار گیرند:
TwitterBootstrapMvc (نسخه سورس باز مطلب جاری است)
Mvc Bootstrap Html Helper Extensions
Bootstrap Helpers (معرفی در اینجا)
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 برا اساس امکانات Bootstrap 3 نوشته شده است و هدفش تسهیل استفاده از Bootstrap 3 در MVC می باشد 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

twitter Bootstrap کتابخانه جامعی از CSSها و JScriptها برای طراحی صفحات وب می‌باشد ک امروزه بسیار مقبول واقع شده 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
مطلب بسیار مفیدی بود ممنون.
مشکلی که من دارم اینه که اولا از طریق نیوگت قابل نصب نیست یعنی طبق سایت خودشون اسم بکیجشو که مینویسم چیزی پیدا نمیکنه...
منم dll هارو از سایت خودشون دانلود کردم و دستی add refrence کردم... در webconfig قسمت viewها هم دو تگ مربوط را اضاف کردم.

اما در ریزور تا اینجا Bootstrap . Html () را میشناسد اما بعد از پرانتز intelli-sense آن را نمیشناسد. مشکل از کجاست؟ ضمنا نسخه vs من 2013 و از mvc5 استفاده میکنم

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۹/۱۵ ۱۱:۳۳

- نسخه سایت خودش سی روزه هست. باید یک فایل مجوز هم براش 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: ۱۳۹۲/۰۹/۱۶ ۱۹:۳۸

سلام ، از NuGet قابل دریافته ولی نیاز به لایسنز داره که باید بخری و نسخه مجانیش فقط سی روزست ، فایل‌های که پیوست کردم این مشکل و نداره و آخرین نسخست ،راه درستش اینه که اول Bootstrap 3 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: ۱۳۹۲/۱۲/۲۳ ۱۷:۴۷

با سلام
آیا با Bootstrap.rtl 3 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 برام Intelisence رو که میاره داخلش BootStrap نداره.

عنوان: **بررسی خطای Circular References در ASP.NET MVC Json Serialization**
نویسنده: مجتبی کاویانی
تاریخ: ۱۳۹۲/۰۹/۱۸ ۰:۰
آدرس: www.dotnettips.info
گروه‌ها: Entity framework, JSON, Serialization, MVC

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 باشید که با خودش کلید خارجی 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 Json Serialization پش فرض ASP.NET MVC 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 آوره 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 بی نام 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 متغییر data ریخته شده و سپس با یک کوئری دیگر که در آن دوباره از پروپرتی items که از نوع کلاس item می‌باشد شی بی نامی ایجاد نموده ایم. با این کار براحتی این خطا رفع می‌گردد.

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۹/۱۸ ۰:۴۹

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 ۱۶:۲

خیر؛ این خطا مربوط به Json Serialization می‌باشد. ProxyCreation برای مباحث Lazy Loading و Change Tracking کاربرد دارد.

نویسنده: Ara
تاریخ: ۱۳۹۲/۰۹/۲۷ ۲۳:۹

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 ۱۸:۵

سلام وخسته 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 بزارید درست میشه

```
GlobalConfiguration.Configuration.Formatters.JsonFormatter.SerializerSettings.ReferenceLoopHandling =
  Newtonsoft.Json.ReferenceLoopHandling.Serialize;
GlobalConfiguration.Configuration.Formatters.JsonFormatter.SerializerSettings.PreserveReferencesHandlin
g =
  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 ۱۸:۱۸

GlobalConfiguration.Configuration.Formatters مربوط به Web API هست. برای MVC باید return Json توکار رو با نمونه Newtonsoft.Json در همه جا تعویض کنید.

نویسنده: محمد
تاریخ: ۱۳۹۳/۰۶/۱۲ ۱۸:۲۸

خیلی ممنون . میشه یه نمونه کد یا سایت یا چیزی برام بزارید که من دقیقا 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 ReferenceLoopHandling.Ignore متد Execute.... اون رو override کردم . جواب هم داد . فقط یه گیری داشت .اونم اینکه من تو مدلم یک فیلدی دارم که از نوع Byte[] هستش . و توش 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 serelize میکنم همه چیزو serelize 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 فیلدهای باینری رو serelize نکنه . ممنون میشم کمکم 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: ۱۳۹۲/۰۹/۱۹ ۲۳:۴۵
آدرس: www.dotnettips.info
برچسب‌ها: MVC, ActionFilters

همانطور که می‌دانید در صورت عدم تعریف صریح layout در یک View، این تعریف از فایل Views_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 اکشن متدهای 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 موردنظرمان است را به خاصیت MasterName اختصاص میدهیم.

عنوان: **حذف فضاهای خالی در خروجی صفحات ASP.NET MVC**
نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۰۹/۲۲ ۲۰:۳۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, Performance, ActionFilters, HTTP Compression

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 صرفا 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 قاعده الگوهای مورد نظر لیست شده و سپس حذف می‌شوند و در نهایت خروجی مورد نظر حاصل خواهد شد. برای راحتی کار و عدم نوشتن 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 روال کار به این صورت نیست و نمی‌توان مانند ASP.NET Webform عمل کرد.

چون در MVC از ViewPage استفاده می‌شود و ما مستقیما 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




            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 deflate فشرده سازی می‌کند. تا اینجای 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 هربار از خیرش گذشتم به این دلایل:
- در مرورگرهای قدیمی گاها 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 علاقمند هستند که doctype ابتدای یک فایل HTML، حتما در یک سطر مجزا ذکر شود.
- زمانیکه شما codeایی 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 GZIP یا Deflate، حداقل کاری را که به خوبی انجام می‌دهند، فشرده سازی فواصل خالی است.

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۲/۰۹/۲۲ ۲۱:۵

بله کاملا حق با شماست و مشکل زمانی زیاد می‌شود که در صفحه کد js 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 بیفتد.
که باید حتی الامکان از این نوع کامنت‌ها استفاده نشود.
در هر صورت از نظر شما متشکرم

نویسنده: میثم هوشمند
تاریخ: ۱۳۹۲/۰۹/۲۳ ۰:۳۱

با سلام
می‌شود این استثناها 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 مثلا می‌شود تگ Pre را لحاظ نکرد:

```
var regex = new Regex(@"(?<=\s)\s+(?![^<>]*</pre>)");
```

عنوان: **جلوگیری از درج صفحات سایت در سایتی دیگر از طریق iframeها**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۲۴ ۸:۰
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, JavaScript, MVC, Security

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 این‌کار 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 جاری، window رویی نبود)، بنابراین صفحه‌ی جاری، داخل iframe قرار گرفته‌است. 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: ۱۳۹۲/۰۹/۲۴ ۹:۴

استفاده از X-FRAME-OPTIONS در هدر درخواست با مقدار SAMEORIGIN با DENY کفایت نمیکنه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۲۴ ۹:۲۰

خوبه ؛ فقط redirect بحث شده در مطلب را نداره

نویسنده: امید قربانی
تاریخ: ۱۳۹۲/۰۹/۲۷ ۹:۱۹

با سلام؛ اگر بخواهیم این موردی 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 ببریمش ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۲۷ ۹:۲۶

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: ۱۳۹۲/۱۲/۲۸ ۲۲:۱۰

سلام
آیا سمت سرور راهی هست که بتوان تشخیص داد صفحاتی که درخواست داده می‌شوند از درون یک Iframe فرخوانی می‌شوند یا خیر(به طور خاص در ASP.NET MVC)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۲۸ ۲۲:۵۹

خیر. راه حل تشخیص سمت سرور، ندارد. هیچ هدر خاصی برای مشخص سازی درخواست اجرای یک صفحه از درون یک IFrame طراحی نشده‌است. حداکثر 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)

عنوان: مشکل اعتبار سنجی jQuery validator در Bootstrap tabs
نویسنده: ابوالفضل رجب پور
تاریخ: ۱۳۹۲/۰۹/۲۴ ۹:۱۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, jQuery, Validation, Twitter Bootstrap

با استفاده از چهارچوب بوت استرپ می‌توان رابطهای کاربر استانداردی ساخت که قبلا دوستان در این سایت مطالبی را در این باب نوشته اند.

در مطلب صفحات مودال در بوت استرپ 3 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 کنترل هایی که در تب‌های دیگر هستند و احیانا 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('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('.form-group').removeClass('has-error').addClass('has-success');
            }
            $(element).trigger('unhighlited');
        }
    });
}
```

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 اعتبارسنجی فیلدهای 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 می‌شوید که **فقط** کنترل‌های تب جاری اعتبارسنجی میشوند و بدون توجه به سایر کنترل هایی که در تب‌های دیگر هستند». این tab جاری منظور همون به ازای هر tab جداگانه هست.

برای نمایش خودکار 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
```

عنوان: افزودن هدرهای Content Security Policy به برنامه‌های ASP.NET
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۲۹ ۱۳:۰
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, MVC, Security

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 تمهیداتی را درنظر نگرفته‌است، خنثی کنند. این قابلیت‌ها به صورت پیش فرض فعال نبوده و نیاز است برنامه نویس صراحتا 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 فعلا 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، مثلا مانند الحاق jQuery به صفحه، نیاز است مسیر آن‌را صراحتا در این هدر ذکر کنید:

```
Content-Security-Policy: script-src 'self' https://youcdn.com
X-WebKit-CSP: script-src 'self' https://yourcdn.com
X-Content-Security-Policy: script-src 'self' https://yourcdn.com
```

علاوه بر آن حتی می‌شود پردازش تمام منابع مورد استفاده را نیز مانند تصاویر، شیوه‌نامه‌ها، فایل‌های فلش و غیره، به دومین جاری محدود کرد:

```
Content-Security-Policy: default-src 'self' https://youcdn.com
X-WebKit-CSP: default-src 'self' https://yourcdn.com
X-Content-Security-Policy: default-src 'self' https://yourcdn.com
```

بدیهی است پس از آشنایی با این مورد، احتمالا در پروژه‌های جدید خود از آن استفاده کنید (چون inline scriptهای فعلی شما را کاملا از کار می‌اندازد).

نحوه‌ی 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 آن‌را 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 ماژول 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 X-Content-Type-Options به nosniff سبب می‌شود تا IE سعی نکند با اجرای یک محتوا سعی در تشخیص mime-type 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/

یک نکته تکمیلی

توصیه شده‌است تا دیگر از روال رویدادگردان PreSendRequestHeaders برای ارسال هدرها استفاده نکنید؛ چون با پردازش‌های غیرهمزمان تداخل ایجاد می‌کند.

نظرات خوانندگان

نویسنده: محمد رعیت پیشه
تاریخ: ۱۳۹۲/۰۹/۳۰ ۹:۲۳

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: ۱۳۹۲/۰۹/۳۰ ۹:۲۷

اجرا نمی‌شود؛ مگر اینکه به هدر ارسالی مانند CDN ذکر شده در متن، آدرس آن اضافه و صراحتا مجوز استفاده از دومینی دیگر توسط برنامه صادر شود.

نویسنده: ناصر طاهری
تاریخ: ۱۳۹۲/۰۹/۳۰ ۱۲:۲۲

یعنی کلیه اسکری‌ت‌ها حتی اسکری‌ت 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: ۱۳۹۲/۰۹/۳۰ ۱۳:۱۸

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: ۱۳۹۲/۰۹/۳۰ ۱۳:۳۷

ممنون از مثالتون. یک پروژه فروشگاه دارم که داخل هر صفحه‌ی آن پر است از اسکریپت‌های به قول شما مدفون شده. حتی فکر پیدا کردن و منتقل کردن هر کدوم از اونها به داخل یک فایل خارجی وحشت آور است.

نویسنده: رسول آذری
تاریخ: ۱۳۹۲/۱۰/۰۴ ۱۳:۴۷

با سلام . تشکر

من با مرورگر ie9 تست کردم . ولی کد اجرا شد. مگه نگفتین 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: ۱۳۹۲/۱۰/۰۴ ۱۴:۱۸

- رفتار IE در مورد X-XSS-Protection متفاوت است. یعنی باید واقعا 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 الحاق کنند »

نویسنده: رسول آذری
تاریخ: ۱۳۹۲/۱۰/۰۴ ۱۴:۳۵

ممنون از پاسخ تون.
یه سری از کدها را میشه توی فایل خارجی قرار داد.
ولی کدهای جاوایی که گریدویو تولید میکنه،چی کار میشه کرد؟
با استفاده از این روش رویدادهایی مثل SelectedIndexChanged در dropdownlist ،
RowCommand در گریدویو ، updatepanel و scriptmanager از کار می‌افته.آیا راهی برای این مشکل وجود داره؟
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۰/۰۴ ۱۵:۲

- به ASP.NET 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 الحاق شده به صفحه نیز میسر است.

عنوان: **امکان اعتبارسنجی با تاخیر در ASP.NET 4.5**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۰/۰۴ ۱:۰
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, MVC, Security

در نگارش‌های قبلی ASP.NET Web forms اگر نیاز به ارسال محتوای HTML ایی وجود داشت، می‌بایستی کل سیستم اعتبارسنجی حداقل یک صفحه را غیرفعال کرد. برای مثال:

```
<%@ Page Language="C#" ValidateRequest="false" %>
```

این نقیصه‌ی همه یا هیچ، در ASP.NET MVC وجود ندارد و می‌توان به ازای یک خاصیت خاص، اعتبارسنجی پیش فرض را با اعمال ویژگی AllowHtml موقتا غیرفعال کرد؛ اما مابقی فیلدها و خاصیت‌های فرم همچنان تحت نظر سیستم اعتبارسنجی‌های ورودی ASP.NET قرار خواهند داشت و به این ترتیب امکان ورود اطلاعات خطرناک، خصوصا 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 ASP.NET 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 خواهید داشت. به این ترتیب دیگر نیازی نخواهید داشت تا اعتبارسنجی کل صفحه را برای دسترسی به یک مقدار خاص غیرفعال نمائید.
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 Idهای واقعی دریافت شده از کاربر مانند ctl00$MainContent$MessageText می‌باشد.

روش دوم، استفاده از خاصیت جدید ValidateRequestMode یک کنترل سمت سرور است و در اینجا نیز می‌توان صرفا اعتبارسنجی یک کنترل را بجای کل صفحه، غیرفعال کرد:

```
<asp:TextBox ID="txtASPNet" 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 فعال سازی اعتبارسنجی با تاخیر نیاز است اصلاح ذیل را نیز به web.config برنامه اعمال کنید (مقدار پیش فرض آن 4 است):

```
<httpRuntime targetFramework="4.5" requestValidationMode="4.5" />
```

عنوان: **معرفی ASP.NET Identity**
نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۰/۱۴ ۹:۴۵
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, Entity framework, MVC, Security, ASP.NET Identity

سیستم ASP.NET Membership بهمراه ASP.NET 2.0 در سال 2005 معرفی شد، و از آن زمان تا بحال تغییرات زیادی در چگونگی مدیریت احزار 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 دهندگان می‌خواهند 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، مثلا 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؛ مثلا Windows Azure Storage Tables. اما در این صورت باید مقادیر زیادی کد بنویسید. مقادیر زیادی هم System.NotImplementedException باید بنویسید، برای متد هایی 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 کامپوننت هایی برای احراز هویت است که شامل سرویس‌های خارجی هم می‌شود (مانند Microsoft Accounts, Facebook, Google, Twitter). همچنین امکان ورود به سیستم توسط حساب‌های کاربری سازمانی (Organizational Accounts) نیز وجود دارد مانند Active Directory و Windows Azure Active Directory. این کتابخانه از JWT، OAuth 2.0 و CORS نیز پشتیبانی می‌کند.

ASP.NET Simple Membership

ASP.NET simple membership به عنوان یک سیستم عضویت، برای فریم ورک Web Pages توسعه داده شد. این سیستم با WebMatrix و Visual Studio 2010 SP1 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. محدودیت هایی نیز وجود دارند:

ذخیره داده‌های سیستم عضویت در بانک‌های اطلاعاتی non-relational مشکل است.
نمی توانید 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

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 داده ای 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

Membership 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 اوقت بصورت زنده! شبکه هایی 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 دلواپسی‌های توسعه دهندگان شده است. در سال 2008 تیم ASP.NET فریم ورک جدیدی را بر اساس الگوی Model-View-Controller (MVC) اضافه کردند. هدف آن کمک به توسعه دهندگان، برای تولید برنامه‌های ASP.NET با قابلیت Unit Testing بهتر بود. توسعه دهندگانی 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 ASP.NET MVC, Web Forms, Web Pages, Web API و SignalR

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. مثلا 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 دیتابیس‌های NoSQL.

تست پذیری

ASP.NET Identity تست پذیری اپلیکیشن وب شما را بیشتر می‌کند. می‌توانید برای تمام قسمت هایی که از ASP.NET Identity استفاده می‌کنند تست بنویسید.

تامین کننده نقش (Role Provider)

تامین کننده ای وجود دارد که به شما امکان محدود کردن سطوح دسترسی بر اساس نقوش 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 دسته ای از اختیارات او شناسایی می‌شود. با استفاده از این روش توسعه دهندگان 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 کنندگان 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.

ASP.NET Identity از OWIN برای ورود/خروج کاربران در سایت استفاده می‌کند. این بدین معنا است که بجای استفاده از Forms Authentication برای تولید یک کوکی، از OWIN CookieAuthentication استفاده می‌شود.

پکیج NuGet

ASP.NET Identity در قالب یک بسته NuGet توزیع می‌شود. این بسته در قالب پروژه‌های ASP.NET MVC, Web Forms و Web API که 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 چابک به توسعه دهندگان عرضه کنند.

شروع کار با ASP.NET Identity

ASP.NET Identity در قالب پروژه‌های ASP.NET MVC, Web Forms, Web API و SPA 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:

Microsoft.AspNet.Identity.EntityFramework این بسته شامل پیاده سازی ASP.NET Identity با Entity Framework می‌شود، که تمام داده‌های مربوطه را در یک دیتابیس SQL Server ذخیره می‌کند.
Microsoft.AspNet.Identity.Core 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 Storage Table و دیتابیس‌های NoSQL.

Microsoft.AspNet.Identity.OWIN 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.

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 متد Register را فراخوانی می‌کند تا حساب کاربری جدیدی با استفاده از ASP.NET Identity API 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 متد 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 ASP.NET Identity و OWIN Cookie Authentication هر دو Claims-based هستند، فریم ورک، انتظار آبجکتی از نوع ClaimsIdentity را خواهد داشت. این آبجکت تمامی اطلاعات لازم برای تشخیص هویت کاربر را در بر دارد. مثلا اینکه کاربر

مورد نظر به چه نقش هایی تعلق دارد؟ و اطلاعاتی از این قبیل. در این مرحله می‌توانید Claimهای بیشتری را به کاربر بیافزایید.

کلیک کردن روی لینک Log off در سایت، اکشن متد 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. متد SignOut همتای متد FormsAuthentication.SignOut است.

کامپوننت‌های 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 بسته‌ها وابستگی هایی هستند که برای استفاده از ASP.NET Identity در اپلیکیشن‌های ASP.NET لازم اند.

دو پکیج دیگر نیز وجود دارند که به آنها اشاره نشد:

Microsoft.Security.Owin.Cookies این بسته امکان استفاده از مدل احراز هویت مبتنی بر کوکی (Cookie-based Authentication) را فراهم می‌کند. مدلی مانند سیستم ASP.NET Forms Authentication.

EntityFramework که نیازی به معرفی ندارد.

مهاجرت از Membership به ASP.NET Identity

تیم ASP.NET و مایکروسافت هنوز راهنمایی رسمی، برای این مقوله ارائه نکرده اند. گرچه پست‌های وبلاگ‌ها و منابع مختلفی وجود دارند که از جنبه‌های مختلفی به این مقوله پرداخته اند. امیدواریم تا در آینده نزدیک مایکروسافت راهنمایی‌های 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. تفاوت‌های فنی و ساختاری زیادی وجود دارند، مثلا الگوی دیتابیس‌ها 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.

قدم‌های بعدی Create an ASP.NET MVC 5 App with Facebook and Google OAuth2 and OpenID Sign-on

در این مقاله خواهید دید چگونه اطلاعات پروفایل را اضافه کنید و چطور از ASP.NET Identity برای احراز هویت کاربران توسط Facebook و Google استفاده کنید. Deploy a Secure ASP.NET MVC app with Membership, OAuth, and SQL Database to a 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 رایج نمایش داده شده.

نظرات خوانندگان

نویسنده: 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 رو نداره سراغ دارین؟ لینکهای بالا همشون موردی هستن

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۰/۱۴ ۱۲:۳۰

مباحث ابتدایی Forms Authentication مربوط است به ASP.NET 1.x؛ دوره‌ای در این مورد . همچنین در MVC هم قابل استفاده است (^ و ^). مباحث membership هم مربوط است به ASP.NET 2.x. کتابی در این مورد .

نویسنده: 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؟

نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۰/۱۴ ۲۱:۴۴

تا بحال با کتاب یا دوره جامعی درباره ASP.NET Identity مواجه نشدم، اگر منبع مناسبی پیدا کنم به اشتراک میذارم. در چند پست آتی بیشتر درباره این فریم ورک صحبت خواهم کرد و مثال هایی عملی نیز در نظر خواهم گرفت

نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۰/۱۴ ۲۳:۲۷

سوالتون رو دقیقا متوجه نشدم. از خاصیت Authorize میتونید استفاده کنید، که قابل اعمال بر روی تک تک اکشن متدها و یا کل کنترلر است. خاصیت AllowAnonymous برای دسترسی عمومی استفاده می‌شود. برای اطلاعات بیشتر درباره نحوه استفاده از ASP.NET Identity و ساختار کلی OWIN 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() { }
}
```

نویسنده: 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 متد 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 براش زحمت می‌کشید.

اگر ممکنه کمی در مورد ساختار Identity و کلاس‌ها و اینترفیس‌ها و نحوه کار با اونها بصورتی که بتونیم خودمون هم Custom Implement اش رو انجام بدیم توضیح بدید.

اینکه اینترفیس هایی چون IUser و IUserStore و IUserPasswordStore و IUserSecurityStampStore و در طرف دیگه UserManager و UserStore و IdentityUser چی هستند و با چه هدفی تعریف شدند و قراره چیکار کنند و در آخر برای اینکه 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؟

ممنون

نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۰/۲۳ ۲۰:۴

در پست‌های آتی پیاده سازی یک تامین کننده (Provider) برای MySQL رو 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: دقیقا متوجه نشدم منظورتون کدوم الگوها است، اما چند نکته تکمیلی:
بصورت پیش فرض وقتی ASP.NET Identity به پروژه اضافه میشه کلاسی بنام ApplicationDbContext ایجاد میشه که بعنوان DbContext پیش فرض برای دیتابیس عضویت استفاده خواهد شد. اگر موجودیت جدیدی برای پروفایل کاربران بسازید باید بعنوان یک DbSet<T> به این کلاس افزوده بشه. اگر نیازی به تفکیک دیتابیس سیستم عضویت و دیتابیس اپلیکیشن نیست، بهتره از یک DbContext استفاده کنید. لایه بندی مدل ها، سرویس‌ها و کد دسترسی به داده‌ها هم ساده است چرا که کل سیستم توسط EF Code First مدیریت میشه. بنابراین استفاده از الگوهای رایج مانند تزریق وابستگی‌ها و غیره مشابه دیگر سناریوهای EF Code First است.

نویسنده: کامران سادین
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 می‌تونید همچین کاری بکنید اما توصیه نمیشه.
سیستم Identity اکثر عملیاتش رو بصورت Async انجام میده که نیاز به NET 4.5. داره. دلایل دیگه ای هم وجود داره که اگر یک جستجوی ساده در اینترنت بکنید مطالب خوبی در این باره پیدا می‌کنید، مثلا 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 می‌کنه و DbContext هم مربوط به این کلاس میشه، 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 میشه 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 می‌تونن تو لایه دیگری مثل Domain Models تعریف بشن.

نویسنده: رضا گرمارودی
تاریخ: ۱۳۹۲/۱۲/۰۳ ۹:۲۱

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: ۱۳۹۲/۱۲/۰۳ ۹:۳۶

این پروژه سورس باز هست. مشکلات آن‌را برای رفع در اینجا گزارش کنید. نحوه‌ی گزارش مشکل هم باید کمی فنی باشد. حداقل جزئیات exception و stack trace آن باید گزارش شوند.

نویسنده: احمد
تاریخ: ۱۳۹۳/۰۲/۲۳ ۱۰:۲۸

سلام
آیا در پروژه‌های windows application که از wcf استفاده میکنند هم میتوانیم از این سیستم استفاده کنیم.؟

نویسنده: داریوش حمیدی
تاریخ: ۱۳۹۳/۰۷/۱۸ ۲۰:۴

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: ۱۳۹۳/۰۹/۲۹ ۱۳:۴۶

« اعمال تزریق وابستگی‌ها به مثال رسمی ASP.NET Identity »

عنوان: **متدهای احراز هویت در VS 2013**
نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۰/۱۵ ۲:۱۵
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, MVC, Authentication, ASP.NET Identity, Security

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 نخواهند 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. ASP.NET Identity 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 بعدا آنرا بر روی SQL Server یا Windows Azure SQL Database نیز منتشر کنید.

این قابلیت‌ها در Visual Studio 2013 در نسخه قبلی نیز وجود داشتند، اما کد سیستم عضویت آن مجددا بازنویسی شده‌است. این بازنویسی دارای مزایای زیر است:

سیستم عضویت جدید بجای استفاده از ماژول 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 پروفایل هایشان 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 درباره‌ی 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 **Organizational Accounts** را انتخاب کنید پروژه ایجاد شده برای استفاده از Windows Identity Foundation (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 گزینه‌ی **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 اینترانتی (Intranet) استفاده خواهند شد.

یک راه دیگر انتخاب گزینه On-Premises زیر شاخه Organizational Accounts است. این گزینه بجای استفاده از ماژول Windows Authentication از فریم ورک Windows Identity Foundation برای احراز هویت استفاده می‌کند. انجام چند مرحله دستی برای پیکربندی این گزینه لازم است، اما WIF امکاناتی را عرضه می‌کند که در ماژول احراز هویت ویندوز وجود ندارند. برای مثال، هنگام استفاده از WIF می‌توانید تنظیمات لازم را در Active Directory انجام دهید تا قادر به واکشی اطلاعات پوشه‌ها باشید (directory data querying).

گزینه‌های احراز هویت Organizational Accounts

دیالوگ **Configure Authentication** گزینه‌های متعددی برای احراز هویت توسط (Windows Azure Active Directory including Office 365) و Windows Server 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 WAAD را انتخاب کنید، باید اطلاعات هویتی (Credentials) یک مدیر کل را وارد کنید. برای نسخه نهایی Visual Studio 2013 برنامه هایی وجود دارد تا دیگر نیازی نباشد چنین مراحلی را تکمیل کنید. در این صورت ویژوال استودیو تنظیماتی را نمایش خواهد داد که یک مدیر می‌تواند بعدا از آنها استفاده کند تا اپلیکیشن را بصورت دستی در WAAD 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 داریم که برای کارمندان این سازمان از طریق company.onmicrosoft.com قابل دسترسی خواهد بود. نمی‌توانید WAAD را طوری پیکربندی کنید که کاربران tenantهای دیگر نیز به اپلیکیشن شما دسترسی داشته باشند.

Domain

نام دامنه‌ای در WAAD که می‌خواهید اپلیکیشن را برای آن پیکربندی کنید، مثلا company.onmicrosoft.com. اگر از custom domain

ها استفاده می‌کنید مانند company.com بجای company.onmicrosoft.com می‌توانید این اطلاعات را اینجا وارد کنید.

سطح دسترسی

اگر اپلیکیشن نیاز به کوئری گرفتن یا بروز رسانی اطلاعات پوشه‌ها (directory information) را توسط Graph API دارد، از گزینه‌های **Single Sign-On, Read Directory Data** و یا **Single Sign-On, Read and Write Directory Data** استفاده کنید. در غیر اینصورت گزینه **Single Sign-On** را رها کنید. برای اطلاعات بیشتر به Application Access Levels و Using the Graph API to Query Windows Azure AD مراجعه کنید.

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 https://contoso.onmicrosoft.com/Example می‌شود. اگر می‌خواهید بصورت دستی این فیلد را مقدار دهی کنید، گزینه **More Options** را انتخاب کنید. این شناسه باید با https:// 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 ببینید 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. مثلا فرض کنید نام پروژه Example است و اپلیکیشنی نیز با شناسه https://contoso.onmicrosoft.com/Example در WAAD وجود دارد. در این صورت اپلیکیشن جدیدی با شناسه ای 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 نهایتا 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 WAAD tenantهای متعددی تعریف شده‌اند. برای مثال، نام سایت contoso.com است و برای کارمندان دو سازمان از طریق آدرس‌های contoso.onmicrosoft.com و fabrikam.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 اینترانتی، 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 پوشه‌ها و کنترل سطوح دسترسی در Active Directory 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.

قدم‌های بعدی

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 برام پیش اومده...
- 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 برناممو 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

عنوان: **توسعه اپلیکیشن های ASP.NET با Windows Azure Active Directory**
نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۰/۱۵ ۵:۱۵
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, MVC, Visual Studio 2013, ASP.NET Identity, Windows Azure Active Directory, Security

ابزار ASP.NET برای Windows Azure Active Directory فعال کردن احراز هویت در وب اپلیکیشن هایی که روی Windows Azure Web Sites 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، اپلیکیشن شما را طوری پیکربندی می‌کند تا تمامی کاربران را با استفاده از یک Windows Azure Active Directory tenant احراز هویت کند.
این مقاله ساختن یک اپلیکیشن ASP.NET را که بر اساس organizational accounts پیکربندی شده و بر روی Windows Azure Active Directory میزبانی می‌شود، مرور می‌کند.
پیش نیاز ها

Visual Studio Express 2013 RC for Web یا Visual Studio 2013 RC

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 Windows Azure Active Directory (AD) انتخاب یا ایجاد کنید. اگر قبلا 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 Directory** 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.

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 متناوب هم نیاز دارند. به مرحله بعد بروید.

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 **Organizational Accounts** را انتخاب کنید. نام دامنه خود را وارد کنید و سپس گزینه **Single Sign On, Read directory data** 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 می‌توانید آنها را ویرایش کنید، مثلا از طریق Windows Azure Portal دامنه‌های 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 می‌کند 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 نمی‌شود؛ چرا که Certificateهای استفاده شده trusted هستند.

با حساب کاربری سازمانی (organizational account) که ایجاد کرده‌اید، وارد شوید.

همانطور که مشاهده می‌کنید هم اکنون به سایت وارد شده اید.

توزیع اپلیکیشن روی Windows Azure

در Windows Azure Portal یک وب سایت را به‌همراه 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.

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 organizational accounts یک متد و نما بنام UserProfile 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.

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 زحمات شما
چند وقت پیش قصد داشتم ثبت نام کنم، زمانی که میخواهید حساب کاربری ایجاد کنید نیاز به شماره موبایل داره تا اجازه بده مراحل ثبت نام کامل بشه ولی اصلا امکان انتخاب کشور ایران وجود ندارد و موفق نشدم

نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۰/۱۶ ۱۳:۱۴

از حساب‌های کاربری مایکروسافت می تونید استفاده کنید.

نویسنده: بهروز
تاریخ: ۱۳۹۲/۱۰/۱۹ ۱۲:۱

ممنون بابت مطلب خوبتون.

لطفا یکم شفاف‌تر بگین که چطور برا تو حساب کاربری ایجاد کنیم؟ من هرکاری میکنم این صفحه میاد!
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 میخواد!

عنوان: **ساخت یک اپلیکیشن ساده ToDo با ASP.NET Identity**
نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۰/۱۹ ۱۳:۳۵
آدرس: www.dotnettips.info
گروه‌ها: Entity framework, Tips, ASP.NET MVC 5, ASP.NET Identity, MVC

یک سناریوی فرضی را در نظر بگیرید. اگر بخواهیم IdentityDbContext و دیگر DbContextهای اپلیکیشن را ادغام کنیم چه باید کرد؟ مثلا یک سیستم وبلاگ 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 سناریو هایی را بررسی می‌کنیم و best practiceهای مربوطه را مرور می‌کنیم.
در این پست یک اپلیکیشن ساده ToDo خواهیم ساخت که امکان تخصیص to-doها 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.

تعریف 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 بسازید. 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 Identity 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 VS 2013 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 بسازید و مدل EF Code First 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.
لطفا 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 MVC 4 هم وجود داشت. برای اطلاعات بیشتر لطفا به این لینک مراجعه کنید.

```
[Authorize]
public class ToDoController : Controller
```

کنترلر ایجاد شده را ویرایش کنید تا کاربران را به ToDoها اختصاص دهد. در این مثال تنها اکشن متدهای 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 ToDoها 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 ToDoهای 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 ToDoها را ببینند

بدین منظور ما یک اکشن متد جدید به کنترل مربوطه اضافه می‌کنیم که تمام ToDoها 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. مثلا در متدی که 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.Models.ToDo>

  @{
    ViewBag.Title = "Index";
  }

  <h2>List of ToDoes for all Users</h2>
  <p>
      Notice that we can see the User info (UserName) and profile info such as HomeTown for the user as
well.
      This was possible because we associated the User object with a ToDo object and hence
      we can get this rich behavior.
  12:  </p>

  <table class="table">
      <tr>
          <th>
              @Html.DisplayNameFor(model => model.Description)
          </th>
          <th>
              @Html.DisplayNameFor(model => model.IsDone)
          </th>
          <th>@Html.DisplayNameFor(model => model.User.UserName)</th>
          <th>@Html.DisplayNameFor(model => model.User.HomeTown)</th>
      </tr>
  25:
  26:      @foreach (var item in Model)
  27:      {
  28:          <tr>
  29:              <td>
  30:                  @Html.DisplayFor(modelItem => item.Description)
  31:              </td>
  32:              <td>
                  @Html.DisplayFor(modelItem => item.IsDone)
              </td>
              <td>
                  @Html.DisplayFor(modelItem => item.User.UserName)
              </td>
              <td>
                  @Html.DisplayFor(modelItem => item.User.HomeTown)
              </td>
          </tr>
      }

  </table>
```

صفحه Layout را بروز رسانی کنید تا به ToDoها 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. دقت داشته باشید که رکوردهایی که کاربران دیگر ثبت کرده اند برای شما نمایش داده نخواهند شد.

مشاهده تمام ToDoها بعنوان مدیر سایت

روی لینک **ToDoes for User in Role Admin** کلیک کنید. در این مرحله باید مجددا 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 ToDoهای ثبت شده توسط تمام کاربران را مشاهده کنید.

نظرات خوانندگان

نویسنده: اس ام
تاریخ: ۱۳۹۳/۰۱/۰۱ ۲۰:۲۴

میشه این پروژه رو برا دانلود بزارید؟ ممنون.

نویسنده: میثم سلیمانی
تاریخ: ۱۳۹۳/۰۴/۲۸ ۱۷:۴۰

```
var currentUser = await manager.FindByIdAsync(User.Identity.GetUserId());
```

این GetUserId() چرا وجود نداره؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۲۸ ۱۷:۴۹

در فضای نام Microsoft.AspNet.Identity تعریف شده.

نویسنده: محمد رحیم پورابراهیم
تاریخ: ۱۳۹۳/۱۱/۲۳ ۱۹:۱۳

سلام
اگر پروژه ای داشته باشیم با ساختار زیر
MyProject
MyProject.Service
MyProject.Model
(MyProject.Data (Entity framework code first
اگر بخوایم تو لایه MyProject از asp .net identity استفاده کنیم، به چه نحوی باید این کار رو انجام داد که ساختار دیتابیس که در لایه دیتا دچار مشکل نشه و جداول مورد نیاز asp .net identity به دیتا بیس اضافه بشه؟

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۱۱/۲۳ ۲۰:۱۷

مطلب و پروژه اعمال تزریق وابستگی‌ها به مثال رسمی ASP.NET Identity رو مطالعه کنید.

عنوان: دریافت اطلاعات بیشتر از Social Provider ها در VS 2013
نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۰/۱۹ ۱۷:۳۵
آدرس: www.dotnettips.info
گروه‌ها: OpenID, ASP.NET MVC 5, ASP.NET Identity, OAuth, MVC, Security

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، مثلا مایکروسافت، گوگل، توییتر و فیسبوک. هنگامی که توسط یکی از این تامین کننده‌ها کاربری را احراز هویت کردید، می‌توانید اطلاعات بیشتری درخواست کنید. مثلا 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 کننده‌ها 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 ASP.NET MVC 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 scope 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 فراخوانی می‌شود. پس از آنکه کاربر احراز هویت شد و به اپلیکیشن سطح دسترسی لازم را اعطا 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 Views/Shared/_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 Claim هایی که توسط OWIN دریافت کرده ایم را می‌خوانیم، و شناسه FacebookAccessToken را در دیتابیس عضویت ASP.NET Identity ذخیره می‌کنیم.

```
//
        // GET: /Account/LinkLoginCallback
        publicasyncTask<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 14-15 شناسه FacebookAccessToken را در دیتابیس ذخیره می‌کند.

StoreFacebookAuthToken تمام اختیارات (claim)های کاربر را از UserIdentity می‌گیرد و Access Token را در قالب یک User Claim در دیتابیس ذخیره می‌کند. اکشن LinkLoginCallback هنگامی فراخوانی می‌شود که کاربر وارد سایت شده و یک تامین 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 FacebookAccessToken را بعنوان 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 = newApplicationUser() { 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]
        publicasyncTask<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", newExternalLoginConfirmationViewModel { Email
= loginInfo.Email });
            }
        }
```

در آخر شناسه FacebookAccessToken را در دیتابیس ASP.NET Identity ذخیره کنید.

```
privateasyncTask 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. http://nuget.org/packages/Facebook

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]
publicasyncTask<ActionResult> FacebookInfo()
{
    var claimsforUser = await UserManager.GetClaimsAsync(User.Identity.GetUserId());
    var access_token = claimsforUser.FirstOrDefault(x => x.Type == "FacebookAccessToken").Value;
    var fb = newFacebookClient(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 Markup 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 مجددا به سایت خودتان باید هدایت شوید.

حال هنگامی که روی لینک FacebookInfo کلیک می‌کنید باید صفحه ای مشابه تصویر زیر ببینید.

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.

نظرات خوانندگان

نویسنده: حامد شیربندی
تاریخ: ۱۳۹۳/۱۱/۲۹ ۱۹:۳

با سلام و تشکر
توی دریافت اطلاعات لیست دوستان از فیسبوک باید دقت داشت که طبق مستندات فیسبوک برای Graph API در اینجا ما فقط به اطلاعات دوستانی دسترسی خواهیم داشت که از Facebook Login استفاده کرده باشند.
البته این نکته رو هم باید اضافه کنم که برای دسترسی به هر اطلاعاتی از اکانت فیسبوک کاربر، باید مجوزش رو هم ارسال کنیم
بنابراین برای دریافت اطلاعات مربوط به لیست دوستان باید دستور زیر رو هم به کلاس startUp اضافه کنیم

```
x.Scope.Add("user_friends");
```

برای اطلاع از نام کل مجوزها و حتی تست اونها برای دریافت اطلاعات اکانت از Graph API Explorer استفاده کنید.

عنوان: **معرفی کتابخانه Postal برای ASP.NET MVC**
نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۰/۲۰ ۰:۱۵
آدرس: www.dotnettips.info
گروه‌ها: Tips, Razor, ASP.Net MVC, MVC, Email, SMTP

Postal کتابخانه ای 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

نحوه استفاده از 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();
  }
}
```

Postal 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

Postal ایمیل‌ها را توسط **SmtpClient** ارسال می‌کند که در فریم ورک 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. مثلا قابلیت

IntelliSense و یا Compile-time error را نخواهید داشت.

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 مثلا *ExampleEmail* نمایی با نام *Example.cshtml* 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.

Postal یک قرارداد بنام **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, Ninject)، آن را طوری پیکربندی کنید تا یک نمونه از IEmailService به کنترلرها تزریق کند. سپس از این سرویس برای ارسال آبجکت‌های ایمیل‌ها استفاده کنید (بجای فراخوانی متد Email.Send()).

```
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 HTML

Postal ارسال ایمیل‌های ساده (plain text) و HTML را بسیار ساده می‌کند.

قدم اول - نمای اصلی را بسازید. این نما headerها را خواهد داشت و نماهای مورد نیاز را هم رفرنس 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 دقت کنید، Example.cshtml به Example.Text.cshtml تغییر یافته. 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>
```

ضمیمه ها

برای افزودن ضمائم خود به ایمیل ها، متد Attach را فراخوانی کنید.

```
dynamic email = new Email("Example");
email.Attach(new Attachment("c:\\attachment.txt"));
email.Send();
```

جاسازی تصاویر در ایمیل ها

Postal یک HTML Helper 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>
```

متد **EmbedImage** تصویر مورد نظر را در ایمیل شما جاسازی می‌کند و توسط یک تگ <img/> آن را رفرنس می‌کند.

```
To: john@example.org
From: app@example.org
Subject: Image

@Html.EmbedImage("~/content/postal.jpg")
```

Postal سعی می‌کند تا نام فایل تصویر را، بر اساس مسیر تقریبی ریشه اپلیکیشن شما تعیین کند.

Postal بیرون از ASP.NET

Postal می‌تواند نماهای ایمیل‌ها را بیرون از فضای ASP.NET رندر کند. مثلا در یک اپلیکیشن کنسول یا یک سرویس ویندوز. این امر توسط یک View Engine سفارشی میسر می‌شود. تنها نماهای 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);
    }
}
```

محدودیت ها: نمی توانید برای نمای ایمیل هایتان از Layoutها استفاده کنید. همچنین در نماهای خود تنها از مدل‌ها (Models) می‌توانید استفاده کنید، و نه ViewBag.

Email Headers: برای در بر داشتن نام، در آدرس ایمیل از فرمت زیر استفاده کنید.

```
To: John Smith <john@example.org>
```

Multiple Values: برخی از headerها می‌توانند چند مقدار داشته باشند. مثلا 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
```

و یا تکرار header:

```
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. مثلا شاید بخواهید بجای ارسال ایمیل ها، آنها را به یک صف پیام مثل MSMQ انتقال دهید یا بعدا توسط سرویس دیگری ارسال شوند. این آبجکت MailMessage تمامی Header ها، محتوای اصلی ایمیل و ضمائم را در بر خواهد گرفت.

کلاس EmailService در Postal متدی با نام CreateMailMessage 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 لطفا به سایت پروژه Postal مراجعه کنید.

عنوان: استفاده از Web Fonts در اپلیکیشن های ASP.NET MVC
نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۲/۱۰/۲۴ ۲۰:۲۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, Tips, Web fonts

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 Typography بگذرند، و تنها روی المان‌های گرافیکی پایه ای وقت بگذارند که در اپلیکیشن وب نهایتا استفاده خواهند شد.

به همین دلیل است که فریم ورک‌های HTML/CSS هر روزه محبوب‌تر می‌شوند. فریم ورک هایی مانند Twitter Bootstrap, HTML5 Boilerplate, Foundation و غیره. 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-typeهای مناسب، وزن و استایل آنها می‌تواند به القای ایده ها، اهمیت مطالب و احساسات به کاربر کمک کند. همچنین خوانایی و درک مطالب شما هم بهبود می‌یابد، که همگی به کیفیت کلی طراحی شما کمک خواهند کرد.

انتخاب و استفاده از فونت ها

چالش انتخاب فونت برای وب، این است که باید بدانید آنها دقیقا 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 فونت هایی استفاده می‌کردند که احتمال وجود آنها روی بازه گسترده ای 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. مثلا متون مورد نظر را بصورت تصویر درج می‌کردند، یا با استفاده از افزونه ای در مرورگر از فونت خاصی استفاده می‌شد. مشکل عمده این روش‌ها این بود که متون مورد نظر قابل انتخاب و جستجو نبودند.

Web Fonts

یکی از راه حل‌ها برای رفع این محدودیت، توسعه فونت‌های وب بود. فونت‌های وب روی هر پلفترمی 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-faceهای 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 تقریبا 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 تقریبا 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 پارامترهای مختلفی مانند خواص، طبقه بندی، 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 کرده اند).

4. قطعه کد جاوا اسکریپت تولید شده را، در قسمت <head> فایل _Layout.cshtml (Razor) یا Site.Master (ASPX) کپی کنید.

5. CSS Selectorهای 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 می‌دهیم. برای اینکار، خانواده فونت "bistro-script-web" را به تگ body اضافه می‌کنیم.

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 فونت جدید "Bistro Script Web" وب سایت را مانند تصویر زیر رندر خواهد کرد.

همانطور که می‌بینید استفاده از فونت‌های وب بسیار ساده است. اما بهتر است از اینگونه فونت‌ها برای کل سایت استفاده نشود و تنها روی المنت‌های خاصی مانند سر تیترها (h1,h2,etc) اعمال شوند.

نظرات خوانندگان

نویسنده: رضا
تاریخ: ۱۳۹۲/۱۰/۲۴ ۲۰:۴۲

برای فونت فارسی از چی استفاده کنیم؟

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۲۴ ۲۲:۲۱

می‌تونی از فونت‌های فارسی استاندارد سازی شده جدید استفاده کنی.
دایرکتوری فونت‌های فارسی هم هست.

نویسنده: محمد رعیت پیشه
تاریخ: ۱۳۹۲/۱۰/۲۵ ۱۴:۲۰

این سایت هم برای دریافت فونت‌های فارسی یونیکد، جهت نمایش در مرورگرهای مختلف مفید هست.

نویسنده: محمد رعیت پیشه
تاریخ: ۱۳۹۲/۱۰/۲۵ ۱۴:۲۴

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: ۱۳۹۲/۱۰/۲۵ ۱۶:۴۴

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: ۱۳۹۲/۱۰/۳۰ ۱۵:۲۲

از این روش http://css-tricks.com/snippets/css/using-font-face
و از این فونت‌ها http://awebfont.ir

عنوان: **استفاده از Awesomium.NET در برنامه‌های وب**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۵ ۱۵:۰
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, MVC, Awesomium

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 می‌کردم. 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 unmaged code است، عملا پروسه IIS وب 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 آن‌را 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 می‌توانید محصور کننده‌ی دات نتی آن‌را مشاهده کنید. منظور از Packed در اینجا، استفاده از DLLهای فشرده شده‌ی native آن است که در مسیر Awesomium SDK\1.7.3.0\build\bin\packed کپی شده‌اند. بنابراین برای توزیع این نوع برنامه‌ها نیاز است اسمبلی دات نتی Awesomium.Core.dll به همراه دو فایل بومی icudt.dll و awesomium.dll ارائه شوند.

تهیه تصاویر سایت‌ها به کمک Awesomium.NET

پس از نصب Awesomium اگر به مسیر Documents\Awesomium SDK 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 نهایتا می‌توانیم سطح این 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 خطا مشخص شد که:

الف) WebCore.Shutdown فقط باید در پایان کار یک برنامه فراخوانی شود. یعنی اصلا 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 خطا:

```
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، عملا 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 نهایتا 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 صف در یک ترد مجزا مدام در حال بررسی است. اگر مقداری به آن اضافه شده‌است، از صف خارج شده و پردازش خواهد شد.

نمونه‌ی استفاده از آن، در سمت یک برنامه‌ی وب نیز به صورت زیر است. ابتدا ماژول تهیه شده باید در برنامه ثبت شود:

```
<?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(HttpRuntime.AppDomainAppPath, "App_Data\\Thumbnails\\" + host +
".jpg"),
                TempDir = Path.Combine(HttpRuntime.AppDomainAppPath, "App_Data\\Temp")
            });
            lblInfo.Text = "Please wait. Your request will be served shortly.";
        }
```

در اینجا کاربر درخواست خود را در صف قرار می‌دهد. پس از مدتی کار آن در WorkerThread ماژول 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. Awesomium 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: ۱۳۹۲/۱۱/۱۶ ۱۹:۱۳

پروژه فوق در ویژوال استودیو 2012 باز نمی‌شود پیغام عدم سازگاری می‌دهد. آیا راهنمایی می‌توانید بکنید
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۶ ۱۹:۲۳

فایل‌های csproj و sln 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: ۱۳۹۳/۰۱/۰۵ ۲۰:۳

```
Hi Vahid
```

```
You have done an excellent job. But I have one query after generating one image the second one doesnot
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: ۱۳۹۳/۰۱/۰۵ ۲۰:۵۷

```
Remove the AwesomiumModule->Dispose method's body. It's not necessary
```

عنوان: **خواندن اطلاعات از سرور و نمایش آن توسط Angular در ASP.NET MVC**
نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۲/۱۱/۱۸ ۱۶:۴۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, AngularJS

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 می‌دانید برای نمایش تعدادی از اشیاء در انگولار می‌توان 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>
                <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>
```

در کد فوق توسط ویژگی ng-init می‌توانیم داده هایمان را Initialize کنیم و در نهایت توسط ویژگی ng-repeat می‌توانیم داده هایمان را در صفحه نمایش دهیم. این ویژگی دقیقاً مانند یک حلقه foreach عمل میکند؛ مثلا 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 میخواهیم این اطلاعات را از سمت سرور بخوانیم و به صورت فوق نمایش دهیم. ابتدا مدل مان را به این صورت تعریف می‌کنیم :

```
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 می‌کنیم (جهت سهولت دموی کار) و به view مورد نظر پاس می‌دهیم. البته شما می‌توانید این اطلاعات را از دیتابیس بخوانید؛ روال کار فرقی نمی‌کند:

```
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 متد SerializeObject تعیین کرده‌ایم:

```
var settings = new JsonSerializerSettings { ContractResolver=new
CamelCasePropertyNamesContractResolver()};
return JsonConvert.SerializeObject(products,Formatting.None,settings);
```

view را نیز به این صورت تغییر می‌دهیم :

```
@model string
<div ng-init="products = @Model">
    <div>
        <div>
            <table>
                <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 :

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 واکشی اطلاعات توسط $http مشکلم حل میشه؟
یا میشه یک view در انگولار کش نشود و به هر با route به اون view مورد نظر از سرور فراخوانی بشه

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۳/۰۱/۱۵ ۱۱:۵۲

البته می‌تونید 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 درپرستی
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 مجددا 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 .

عنوان: **آموزش Backload (آپلود چندین فایل به طور همزمان با آجاکس)**
نویسنده: شقایق اشتری
تاریخ: ۱۳۹۲/۱۱/۲۵ ۱۴:۳۵
آدرس: www.dotnettips.info
گروه‌ها: MVC

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 گوشه‌ی بالا سمت راست، کلمه‌ی Backload را تایپ می‌کنیم در نتایج Search دو مورد زیر را Install می‌کنید :

* JQuery File Upload Plugin * Backload. A Proffesional 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:

Search Solution Explorer (Ctrl+;)
Solution 'Backload' (1 project)
Backload
Properties
References
App_Data
App_Start
BackloadConfig.cs
BundleConfig.cs
FilterConfig.cs
RouteConfig.cs
WebApiConfig.cs
Content
Backload
FileUpload
themes
Site.css
Controllers
BackloadDemoController.cs
HomeController.cs
Models
Scripts
FileUpload
_references.js
jquery-1.7.1.intellisense.js
jquery-1.7.1.js

modernizr-2.5.3.js
Views
BackloadDemo
Shared
_ViewStart.cshtml
Web.config
Global.asax
packages.config
Web.Backload.Default.config
Web.Backload.xsd
Web.config

تا اینجا ما ابزار Backload را بر روی پروژه‌ی خود نصب کردیم. همانطور که می‌بینید Backload یک Demo به پروژه اضافه کرده که شامل Controller ، View و ... می‌باشد. اکنون پروژه را اجرا کنید و با FileUpload کار کنید. البته توجه داشته باشید که url صفحه‌ای که FileUpload در آن قرار دارد به این صورت است : localhost:PortNumber/BackloadDemo/Index

***نکاتی که باید بدانید:**

عکس هایی 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 آن‌را باز کنید. این View در پروژه، در مسیر Views/ BackloadDemo/Index قرار دارد .

در داخل تگ form یک Hidden Field با نام objectContext اضافه می‌کنید. Value ایی 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 باFile Upload آپلود می‌کنیم، ایجاد می‌شود.

```
orm 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">
```

چنانچه بخواهید در پوشه‌ای که خودتان ایجاد کردید (که در این مثال 02-2014 می‌باشد) پوشه‌های متعدد دیگری هم داشته باشید Hidden Field 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 -
```

اکنون اگر فایل جدیدی را آپلود کنید در مسیر

2014-02/User200/Id20121/Images

ذخیره می‌شود . یعنی بین نام هر پوشه از سمی 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 بخوایم عکس‌های موجود در پوشه‌ی دیگری را نمایش دهیم باید چه کار کنیم؟!
گاهی اوقات نیاز داریم که محتویات 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 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 Backload.Demo.js که در مسیر Scripts/Fileupload هست را باز کنید و url موجود در آنرا مانند عکس زیر تغییر دهید :

```
backload.demo.js
/*
 * 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 مثلا 5 فایل برای شما در گرید ویو مربوط به FileUpload نمایش داده شود، به تابع handler_GetFilesRequestFinished می‌روید. متغیر limit مشخص می‌کند که 5 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 پوشه‌ها می‌باشد. بار دیگر پروژه را اجرا کنید. با استفاده از ابزاری مثل FireBug کدهای html صفحه‌ی خود را ببینید. Hidden Field ایی با نام objectContext را پیدا کنید و Value آنرا به test تغییر دهید. فایلی را آپلود کنید حالا به پوشه‌ی Files موجود در ریشه‌ی پروژه بروید. می‌بینید که پوشه‌ای با نام testایجاد 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 خوبم "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 ی jQuery File Upload استفاده می‌کنه. به نظرم کسی نحوه‌ی استفاده از این کتاب خانه را با توجه به sampleهای خود سایتش یاد بگیره خیلی بهتره. نسخه‌ی angularjs هم داره.

نویسنده: شقایق اشتری
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 گذاشتم.
مورد بعدی اینکه متاسفانه دستوراتی مثل e.param.Searchpath را در مثال‌های آماده‌ی 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 پوشه استفاده می‌شود اگر یک کاربر سایت ما مقدار value این hidden field را تغییر دهد اسم پوشه به کل تغییر می‌کند که در مثال‌های آماده‌ی 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 اشتری
تاریخ: ۱۳۹۳/۰۱/۲۴ ۱۳:۴۹

نکته ای 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 میشد.

عنوان: **یافتن اکشن متدهای Post ایی در ASP.NET MVC که فیلتر CSRF ندارند**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۰۸ ۰:۱۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, Security

روش مرسوم مقابله با حملات CSRF در ASP.NET MVC، استفاده از فیلتر امنیتی ValidateAntiForgeryToken بر روی اکشن متدهایی از نوع Post است و سپس فراخوانی Html.AntiForgeryToken در View متناظر.
با بالا رفتن تعداد اکشن متدهای یک پروژه، ممکن است استفاده از ValidateAntiForgeryToken فراموش شود. در ادامه مثالی را ملاحظه می‌کنید که یک پروژه‌ی ASP.NET MVC را جهت یافتن اکشن متدهای Post ایی که فیلتر ValidateAntiForgeryToken ندارند، اسکن می‌کند:

```
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 متدهای کنترلر جاری
                var actionMethods = controller.GetMethods(BindingFlags.Public | BindingFlags.Instance |
BindingFlags.DeclaredOnly)
                                              .Where(method =>
typeof(ActionResult).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 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)، جدا خواهند شد.
در ادامه در این کنترلرها، متدهایی را بررسی خواهیم کرد که دارای خروجی از نوع ActionResult باشند (فقط اکشن متدها مدنظر هستند). اگر این اکشن متد یافت شده دارای ویژگی HttpPost بود و همچنین ValidateAntiForgeryToken نداشت، یعنی یک مشکل امنیتی که باید برطرف شود.

عنوان: **ایجاد سیستم وضعیت آب و هوا مانند گوگل (بخش اول)**
نویسنده: شقایق اشتری
تاریخ: ۱۳۹۲/۱۲/۱۰ ۱۱:۳۰
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, C#, ASP.Net MVC, MVC, xml

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 نوشتم ولی شما می‌توانید با اندک تغییراتی در کدها، آنرا در ASP.NET وب فرمز نیز استفاده کنید. برای گرفتن آب و هوای هر شهر از Rssهای اعلام وضعیت آب و هوای یاهو استفاده می‌کنم و توضیح خواهم داد که چگونه با Rss آن کار کنید.
Rss 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 می‌خواهم کوئری استرینگ‌های این لینک را برای شما توضیح دهم. هر شهری بر روی کره‌ی زمین یک WOEID یکتا و منحصر بفرد دارد که شما به پارامتر w عدد WOEID شهر موردنظر خود را می‌دهید. بعد از مقداردهی پارامتر w، وقتی این لینک را در آدرس بار مرورگر خود می‌زنید، RSS مربوط به آب و هوای آن شهر را به شما می‌دهد. مثلا WOEID تهران عدد 28350859 می‌باشد.
و این لینک http://weather.yahooapis.com/forecastrss?w=28350859&u=c 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 مد نظر را بر اساس سانتیگراد می‌خواهید.
* اگر مقدار پارامتر u برابر f باشد، یعنی شما دمای آب و هوای آن شهر مورد نظر 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 DropDown استفاده کردم که کاربر با انتخاب هر شهر از DropDown، آب و هوای آن شهر را مشاهده می‌کند.
Action مربوط به صفحه‌ی 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 عهده‌ی خودتان گذاشتم.
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 = "Azarbayejan-e
Sharqhi" };
                            break;
                        }
                    case 2:
                        {
                            rssXml =
XDocument.Load("http://weather.yahooapis.com/forecastrss?w=2345767&u=c");
                            CountryName = new CountryName() { Country = "Iran", City = "Azarbayejan-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 =
```

```
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 می‌دهم : حلقه‌ی for 0 تا 4 (که در کد بالا مشاهده می‌کنید)یعنی اطلاعات 4 روز آینده را برایم برگردان.
من تگ‌های High ، Low ، Day ، Code و text فایل RSS را در این حلقه For می‌خوانم که البته مقادیر این 4 روز را در لیستی اضافه می‌کنم که نوع این لیست هم از نوع YahooWeatherRssItem می‌باشد. من این کلاس را در فایل ضمیمه قرار دادم. اکنون هر کدام از این تگ‌ها را برایتان توضیح می‌دهم:

code : هر آب و هوا 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 می‌دهد .
High: حداکثر دمای آن روز را به می‌دهد .
day: نام روز از هفته را بر می‌گرداند مثلا شنبه ، یکشنبه و ....
text: که توضیحاتی می‌دهد مثلا اگر هوا 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 صفحه‌ی 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 _Weather که Partial 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 هوای یاهو اطلاعات دقیق‌تری را می‌خواهید بدانید به این لینک بروید.
در آموزش بعدی قصد دارم برایتان این بخش را توضیح دهم که بر اساس IP بازدید کننده سایت شما، اطلاعات آب و هوایی 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

ممنون از شما. چند نکته جزئی در مورد بهبود کیفیت کدها. در Viewهای 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 switch 1 و 2 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 اشتری
تاریخ: ۱۳۹۲/۱۲/۱۲ ۸:۳۹

ممنونم از شما که مشکل کد نویسی من را گفتید . extension methodها منظورتون همون helperها هست ؟ لینکی دارید که در مورد این extension methodها آموزشی داده باشد چون من می‌خوام کدهامو تا جایی که می‌شود بهینه بنویسم .

نویسنده: شاهین کیاست
تاریخ: ۱۳۹۲/۱۲/۱۲ ۹:۱۵

یک مثال در اینجا
قسمت آخر (**Ifهای شرطی را در Viewها را در متدهای کمکی کپسوله کنید)**

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۱۲ ۹:۵۱

قسمت switch ایی را که در View نوشتید، تبدیل کنید به یک متد کمکی در کلاسی خارج از View: (مهم نیست متد الحاقی 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 می‌شود.

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

```
select * from weather.forecast where woeid in (select woeid from geo.placefinder where text="CityName")
```

نویسنده: شقایق اشتری
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 method 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 یی نباشد ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۱۲ ۱۰:۴۱

- نه الزاما. 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 مستقیما 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
ممنون از مقاله خوبتون
امکانش هست سورس کد webForm ش رو هم بذارید واسه دانلود؟
ممنون

نویسنده: علی
تاریخ: ۱۳۹۲/۱۲/۲۱ ۱۰:۸

اینجا یک مثال از نحوه کار با YQL رو توضیح داده

نویسنده: ع پارسایی
تاریخ: ۱۳۹۳/۱۱/۱۸ ۱۶:۳۶

با سلام
امکان داره بگید در اکشن، قسمت ViewBag.ProvinceList = _RPosition.Positions چه کار میکنه و RPosition کارش چیه؟

نویسنده: ع پارسایی
تاریخ: ۱۳۹۳/۱۱/۲۱ ۱۹:۳۴

با سلام؛ در مطلب گفته شده برای گرفتن WOEID شهرها به فلان سایت بروید و هر کشوری رو بخواییم باید بریم و دستی واردش اما من دیتابیس یا هر چیز دیگری رو میخوام که قابلیت این رو داشته باشه که کاربر با وارد کردن کشور و شهر بتونه آب و هوا رو

ببینه. شما میتونید راهنمایی ام کنید که چه کار باید بکنم و یا منبعی برای دیتابیس مختصات جغرافیایی کشورها وجود داره؟ ممنون میشم کمکم کنید. با تشکر

نویسنده: شقایق اشتری
تاریخ: ۱۳۹۳/۱۱/۲۶ ۲:۱۳

سلام ، نام استان‌ها را در ViewBag قرار دادم.

نویسنده: شقایق اشتری
تاریخ: ۱۳۹۳/۱۱/۲۶ ۲:۱۴

به دو لینک زیر نگاهی بیندازید:

https://developer.yahoo.com/yql/

http://dev.maxmind.com/geoip/legacy/geolite/

عنوان: **استفاده از F# در پروژه های MVC4**
نویسنده: مسعود پاکدل
تاریخ: ۱۳۹۲/۱۲/۱۱ ۹:۵
آدرس: www.dotnettips.info
گروه‌ها: #MVC, F

در این پست با روش پیاده سازی پروژه‌های WPF با استفاده از F# آشنا شدید. قصد دارم در طی چند پست روش پیاده سازی پروژه‌های Asp.Net MVC 4 با استفاده از F# را شرح دهم.
»اگر با F# آشنایی ندارید می‌توانید از اینجا شروع کنید.
به صورت کلی برای استفاده گسترده از F# در پروژه‌های وب نیاز به یک سری templateهای آماده داریم در غیر این صورت کار کمی سخت خواهد شد. به تصویر زیر دقت نمایید:

واضح است که با توجه به تصویر بالا کنترلرها و البته مدل‌های app و هر آنچه که سمت سرور به آن نیاز است باید با استفاده از F# پیاده سازی شوند. اما هنگامی که کنترلرها با استفاده از 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 بر خلاف حالت پیش فرض C# آن که در قالب یک پروژه ارائه می‌شود در این جا 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 FSharp MVC 4 را جستجو کنید.

بعد از انتخاب نام پروژه و کلیک بر روی Ok (و البته دانلود حدود ده MB اطلاعات) صفحه زیر نمایان می‌شود. 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 Web Api Project را انتخاب کنید در غیر این صورت گزینه Empty 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 می‌شود. همانند تصویر زیر:

بررسی تغییرات

در یک نگاه به راحتی می‌توان 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 می‌دهد فراخواهی متد Start در فایل Global متناظر در پروژه F# است. اگر به قسمت referenceهای 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 پروژه‌های MVC بر اساس زبان C#، کلاسی به نام RouteConfig وجود دارد که در فایل Global.asax.cs فراخوانی می‌شود:

```
RouteConfig.RegisterRoutes(RouteTable.Routes);
```

و کدهای مورد نظر جهت تعیین الگوی مسیر یابی کنترلرها و درخواست‌ها در کلاس RouteConfig نیز به صورت زیر می‌باشد:

```
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 بر خلاف C#، تعریف و اجرای سیستم مسیر یابی در یک فایل انجام می‌شود. در بخش اول ابتدا یک type تعریف می‌شود که معادل با تعیین Routing در زبان C# است. علاوه بر آن متد RegisterRoutes جهت تعیین و انتساب Route Type تعریف 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 پروژه‌های F# MVC 4 اشنا شدیم و از طرفی مشخص شد که برای پیاده سازی این گونه پروژه‌ها حداقل نیاز به دو پروژه داریم. یک پروژه که از نوع C# است ولی در آن فقط View ها و فایل جاوااسکریپتی و البته Css وجود دارد. از طرف دیگر کنترلرها و مدل‌ها 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 مدل‌ها آشنا خواهیم شد.

عنوان: **توسعه کنترلر و مدل در F# MVC4**
نویسنده: مسعود پاکدل
تاریخ: ۱۳۹۲/۱۲/۱۷ ۲۳:۵
آدرس: www.dotnettips.info
گروه‌ها: #MVC, F

در پست قبلی با F# MVC4 Template آشنا شدید. در این پست به توسعه کنترلر و مدل در قالب مثال خواهم 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>
    </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>
</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 داریم(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 ()View 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 وهله 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
```

Attributeها مدل

اگر همچون پروژه‌های C# قصد دارید با استفاده از Attributeها مدل خود را اعتبارسنجی نمایید می‌توانید به صورت زیر اقدام نمایید:

```
open System.ComponentModel.DataAnnotations
 type Book() = [<Required>] member val Name = "" with get, set
```

هم چنین می‌توان Attributeهای مورد نظر برای مدل EntityFramework را نیز اعمال نمود(نظیر Key):

```
namespace FsWeb.Models
open System
open System.ComponentModel.DataAnnotations
type Book() = [<Key>] member val Id = Guid.NewGuid() with get, set
[<Required>] 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 پروژه‌های وب از زبان طراحان آن بیان شده است.

عنوان: تعامل با پایگاه داده با استفاده از EntityFramework در پروژه های F# MVC 4
نویسنده: مسعود پاکدل
تاریخ: ۱۳۹۲/۱۲/۱۸ ۰:۵
آدرس: www.dotnettips.info
گروه‌ها: #Entity framework, MVC, F

در پست‌های قبلی (^ و ^) با template و ساخت کنترلر و مدل در پروژه‌های F# MVC آشنا شدید. در این پست به طراحی Repository با استفاده از EntityFramework خواهیم پرداخت. در ادامه مثال قبل، برای تامین داده‌های مورد نیاز کنترلرها و نمایش آن‌ها در View نیاز به تعامل با پایگاه داده وجود دارد. در نتیجه با استفاده از الگوی Repository، داده‌های مورد نظر را تامین خواهیم کرد. به صورت پیش فرض با نصب Template جاری (F# MVC4) تمامی اسمبلی‌های مورد نیاز برای استفاده از در EF در پروژه‌های F# نیز نصب می‌شود.

پیاده سازی DbContext مورد نیاز

برای ساخت DbContext می‌توان 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 می‌کنید با ارث بری از کلاس DbContext و پاس دادن ConnectionString یا نام آن در فایل app.config، به راحتی FsMVCAppEntities ساخته می‌شود که معادل DbContext پروژه مورد نظر است. با استفاده از دستور do متد SetInitializer برای عملیات migration فراخوانی می‌شود. در پایان نیز یک DbSet به نام Books 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 می‌توانید 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 وهله سازی می‌شود. نکته مهم این است به جای شناسه let از شناسه use استفاده کردم. شناسه use دقیقا معال دستور using(){} در C# است. بعد از اتمام عملیات شی مورد نظر Dispose خواهد شد.
در بخش بعدی یک کوئری از DbSet مورد نظر گرفته می‌شود. این روش Query گرفتن در F# 3.0 مطرح شده است. در نتیجه در نسخه‌های قبلی آن (F# 2.0) اجرای این کوئری باعث خطا می‌شود. اگر قصد دارید با استفاده از F# 2.0 کوئری‌های 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 F# 2.0 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 وهله 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

نویسنده: 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 میتونه مفید باشه

اگه ممکنه در مورد صورت مسئله و راه حل‌های ارائه شده با F# بیشتر بنویسید تا بیشتر مورد استفاده قرار بگیره ،و خیلی کاربردی‌تر موضوع رو ببینیم

ممنون

عنوان: **انجام کارهای زمانبندی شده در برنامه‌های ASP.NET توسط DNT Scheduler**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۲۶ ۱۱:۴۰
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, MVC, Schedule

اگر به دو مطلب استفاده از Quartz.Net (^ و ^) و خصوصا نظرات آن دقت کرده باشید به این نتیجه خواهید رسید که ... این کتابخانه‌ی در اصل جاوایی گنگ طراحی شده‌است. در سایت جاری برای انجام کارهای زمانبندی شده (مانند ارسال ایمیل‌های روزانه خلاصه مطالب، تهیه خروجی PDF و XML سایت، تبدیل پیش نویس‌ها به مطالب، بازسازی ایندکس‌های جستجو و امثال آن) از یک Thread timer استفاده می‌شود که حجم نهایی کتابخانه‌ی محصور کننده و مدیریت کننده‌ی وظایف آن جمعا 8 کیلوبایت است؛ متشکل از ... سه کلاس. در ادامه کدهای کامل و نحوه‌ی 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>
        /// اگر چند جاب در یک زمان مشخص داشتید، این 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.
- متد RunAt ثانیه‌ای یکبار فراخوانی می‌شود (بنابراین بررسی now.Second 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 تصمیم گیری در مورد زمان اجرای کارها مفید نیست.
در متد 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 دهی می‌شود. این کتابخانه قادر است زمان خاموش شدن پروسه‌ی فعلی IIS را تشخیص داده و خاصیت IsShuttingDown را true 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 تصمیم گیری کرد. خاصیت ScheduledTasksCoordinator.Current.ScheduledTasks، لیست وظایف تعریف شده را در اختیار شما قرار می‌دهد.
- در متد Run، منطق وظیفه‌ی تعریف شده را باید مشخص کرد. برای مثال ارسال ایمیل یا تهیه‌ی بک آپ.
- Name 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 مثلا 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 وهله‌ی 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، کار Thread timer برنامه شروع می‌شود.
- اگر در حین اجرای متد Run، استثنایی رخ دهد، آن‌را توسط یک Action delegate به نام ScheduledTasksCoordinator.Current.OnUnexpectedException می‌توانید دریافت کنید. کتابخانه‌ی DNT Scheduler 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 پروسه‌ی 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 بیدار کردن مجدد برنامه استفاده کرد.

استفاده از کلاس ScheduledTasksRegistry تعریف شده

پس از اینکه کلاس ScheduledTasksRegistry را تعریف کردیم، نیاز است آن‌را 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 پایان کار برنامه جهت پاکسازی منابع باید فراخوانی گردد.

همچنین در اینجا با فراخوانی ScheduledTasksRegistry.WakeUp، می‌توانید برنامه را مجددا زنده کنید! IIS مجاز است یک سایت ASP.NET را پس از مثلا 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 فراخوانی مجدد تعریف 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، مجددا برنامه توسط IIS بارگذاری شده و اجرا می‌گردد. به این ترتیب وظایف تعریف شده، در طول یک روز بدون مشکل کار خواهند کرد.

گزارشگیری از وظایف تعریف شده

برای دسترسی به کلیه وظایف تعریف 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 می‌تونه مثلا ارسال یک ایمیل یک هفته قبل از اتمام زمان شارژ کاربری باشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۲۶ ۱۹:۳۷

بله. می‌توانید یک وظیفه‌ی جدید تعریف کنید که هر شب ساعت مثلا 11 و 15 دقیقه اجرا شود (نحوه‌ی تعریف متد RunAt). سپس در متد Run آن یک کوئری از دیتابیس گرفته، لیست موارد مدنظر را واکشی کرده و به آن‌ها ایمیل بزنید.

نویسنده: پیمان مهربانی
تاریخ: ۱۳۹۲/۱۲/۲۷ ۱۱:۴۷

کتابخانه سبکی بود اما نگرانی هایی مانند پیش آمدن همزمانی در اجرای وظایف را دارم.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. 1و 2 و 3 و الی آخر.
- «حتی می‌توان از Scheduler خود ویندوز و یا Jonهای SQL Server هم بهره برد».
بله. به شرطی‌که 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. ضمنا برای ASP.NET نوشته شده است و این قابلیت را دارد که به شما اعلام کند مثلا تا 30 ثانیه دیگر برنامه از سرور unload می‌شود؛ توسط خاصیت IsShuttingDown. همچنین حق تقدم ترد آن طوری تنظیم شده که سبب اختلال در عملیات و عملکرد متداول سایت نشود.
- «آیا توصیه شده برای استفاده در پروژه‌های بزرگ هست؟»
یک به اشتراک گذاری بود از قسمتی از کدهای زیر ساخت سایت جاری که هم اکنون مورد استفاده است (مقدمه بحث).

نویسنده: سلمان کاظمی
تاریخ: ۱۳۹۲/۱۲/۲۸ ۱:۳۲

سوالی که واسه من پیش اومده 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 بده؟اگه بخوام که در برنامه انجام بشه خوب کجا باید نوشت ؟ یا به عبارتی دیگه باید حتما برنامه اجرا بشه و اگه برنامه اصلا 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: ۱۳۹۳/۰۱/۰۱ ۲۳:۲۳

خیلی خیلی ممنون ... کتابخانه‌ی ساده و مفیدی است.
سوال من اینه که میشه کاری کرد که آدرس روت تعریف شده در فایل کانفیگ جهت بیدار شدن IIS را در کد نوشت 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 ست کنیم !
پی نوشت : من این کار رو در متد Application_End نوشتم و خطای در دسترس نبودن Request رو دریافت کردم ! (هر چند که میدونستم این خطا رو میده !)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۰۲ ۰:۵۱

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: ۱۳۹۳/۰۱/۰۲ ۱۳:۲

خیلی خیلی ممنون :) ... اگه میشه نسخه‌ی nuget کتابخانه را هم بذارید

نویسنده: مهدی پایروند
تاریخ: ۱۳۹۳/۰۱/۲۵ ۱۴:۳۱

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 کار شروع برنامه را خود به خود انجام نمی‌دهد؟

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 Application Initialization Module نیز برای اجرا خودکار برنامه پس از ری‌استارت سرور طراحی شده .

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۱۷ ۹:۶

یک نکته‌ی تکمیلی

در دات نت *4.5.2* ، متدی به نام HostingEnvironment.QueueBackgroundWorkItem اضافه شده‌است تا درخواست اجرای کارهای پس زمینه در ASP.NET به سادگی و همچنین با اطمینان بیشتری قابل انجام باشد.

نویسنده: مهرداد
تاریخ: ۱۳۹۳/۰۲/۱۷ ۱۳:۴۰

یعنی با استفاده از این متد جدیدی 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 متد استفاده کرد ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۱۷ ۱۳:۴۲

فقط قسمت مدیریت تردهای آن با یک سطر QueueBackgroundWorkItem جایگزین می‌شود. مابقی قسمت‌های آن (مانند اینکه یک WorkItem چه زمانی در Queue قرار گیرد) تفاوتی نمی‌کند و مانند قبل است.

نویسنده: مهرداد
تاریخ: ۱۳۹۳/۰۲/۱۷ ۱۴:۱۰

منظور شما این خط هست ؟

```
ScheduledTasksCoordinator.Current.AddScheduledTasks(
                new SendEmailsTask(),
                new DoBackupTask());
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۱۷ ۱۴:۲۳

این‌ها باقی خواهند ماند. قسمت new Thread و مدیریت آن جایگزین می‌شود.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۲۹ ۹:۲۵

یک نکته

مثالی از نحوه‌ی استفاده از متد جدید HostingEnvironment.QueueBackgroundWorkItem
+ یک مثال رسمی

نویسنده: Aria
تاریخ: ۱۳۹۳/۰۳/۱۹ ۲۳:۱۰

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 اونو به صورت پس زمینه در حال اجرا نگه داشت.
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۳/۲۰ ۰:۱۹

- 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، احتمالا 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 متد Start کلاس ScheduledTasksCoordinator بود که تغییر میکنه؟
یعنی فقط متد 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 HostingEnvironment.RegisterObject و IRegisteredObject آن هم باید حذف شوند چون در QueueBackgroundWorkItem وجود دارند و یک CancellationToken را تنظیم می‌کند.
+ زمانیکه از DNTScheduler استفاده می‌کنید، عملا نیازی به QueueBackgroundWorkItem ندارید. چون نکته‌ی HostingEnvironment.RegisterObject و IRegisteredObject در آن لحاظ شده. این نکته که خاموش شدن IIS را گزارش می‌کند، چند سال قبل، توسط یکی از اعضای قبلی تیم ASP.NET منتشر شده بود . دقیقا از همین نکته در QueueBackgroundWorkItem استفاده شده.

به صورت خلاصه، DNTScheduler با دات نت 4 به بعد سازگار است و نکات QueueBackgroundWorkItem دات نت 4.5.2 را به صورت توکار پیاده سازی کرده‌است.

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: ۱۳۹۳/۰۵/۰۹ ۱۰:۱۹

سلام
من برای این سری کارها از ویندوز سرویس استفاده می‌کنم. مثلا ویندوز سرویس من از ساعت 8 صبح شروع به کار می‌کنه و رویدادهایی 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 وهمیشگی 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: ۱۳۹۳/۰۵/۰۹ ۱۱:۳

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 بهت گفتن ما فقط اجازه می‌دیم فایل‌های سایتت رو روی سرور کپی کنی. دسترسی بیشتری بهت نمی‌دیم. اون وقت چکار می‌کنی؟

نویسنده: Mohammad
تاریخ: ۱۳۹۳/۰۵/۱۵ ۲:۵۶

سلام. ممنون.

دوستان برای من وقتی به صورت ساعت می‌دم درست کار نمیکنه.مثلا من می‌خوام در ساعت دو و بیست و چهار دقیقه بامداد یه کاری رو انجام بده اینجوری نوشتم:در صورتی اگه بگم دو دقیقه دو دقیقه 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: ۱۳۹۳/۰۵/۱۵ ۸:۴۶

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. مثلا اگر پس از بررسی متوجه شدید ساعت سرور یک ساعت عقب هست، now.Hour == 2 می‌شود now.Hour == 1 و امثال این نوع محاسبات.

نویسنده: علی
تاریخ: ۱۳۹۳/۰۶/۱۵ ۱۸:۳۴

ممنون. مفید بود. اگه بخواهیم یه کاری مثلا هر سه روز یکبار انجام بشه باید چه جوری زمان رو تعیین کنیم؟ ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۶/۱۵ ۱۹:۵۷

```
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: ۱۳۹۳/۰۶/۱۷ ۲۲:۲۲

یه سوال:اختلاف زمانی ساعت گرینویچ با تهران همیشه 3:30 هست؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۶/۱۷ ۲۳:۸

اگر سرور شما تنظیمات daylight saving time صحیحی داشته باشد، بله. اگر نه، خودتان این یک ساعت تفاوت را 6 ماه یکبار باید محاسبه و اعمال کنید.

نویسنده: آرام
تاریخ: ۱۳۹۳/۰۶/۱۸ ۹:۵۷

سلام.
چرا هیچ ارجاعی به متد Stop در کلاس JobsRunnerTimer وجود نداره؟ برای این کار دلیلی دارید؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۶/۱۸ ۱۰:۱۴

نیازی نیست. چون طول عمر کل این ماژول دقیقا 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 ماژول هم به همین ترتیب. هر زمان که IIS تصمیم به خاتمه‌ی برنامه گرفت، نه این ماژول، هیچ ماژول دیگری هم فرصت مقاومت پیدا نمی‌کند و راسا به همراه AppDomain جاری خاتمه می‌یابد.

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۳/۰۸/۰۷ ۱۸:۴۳

ممنون، عالیه
فقط یه سوال:
چطور می‌تونیم یک Task خاص رو خودمون فراخوانی کنیم، مثلاً بعداً از اینکه کاربر در سایت ثبت‌نام کرد به صورت هفتگی براش ایمیل ارسال بشه یا مواردی از این دست، در مطلب بالا این طوری که من متوجه شدم، توسط دستور ScheduledTasksRegistry.Init(); در Application_Start تمام وظایف به صورت همزمان شروع میشن، در این حالت باید برای هر وظیفه یک کلاس برای ریجستر کردن اون بنویسیم؟ درسته؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۸/۰۷ ۱۸:۴۸

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: ۱۳۹۳/۰۸/۰۷ ۲۳:۵۰

ممنون،
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 6 روز، روزانه کاری را انجام دهد، بعد از پایان روز 6 یعنی در روز هفتم یک وظیفه دیگر شروع خواهد شد(تعداد روزها توسط کاربر انجام می‌گیرد). و در نهایت وظایف یکی پس از دیگری بعد زمان تعیین شده برای آنها متوقف خواهند شد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۸/۰۸ ۰:۱

در متد 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 متد ثانیه‌ای یکبار بررسی می‌شود. بنابراین مباحث caching را جهت واکشی اطلاعات از بانک اطلاعاتی در متد RunAt اعمال کنید تا بی‌جهت رفت و برگشت اضافی صورت نگیرد.

نویسنده: حسن
تاریخ: ۱۳۹۳/۰۸/۰۸ ۱:۴۵

سلام
ایا میشه از این کتابخانه به صورت ثانیه ای 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 !
مثلا اینطوری اگر کار کنه مشکلی پیش نمیاد ؟

```
return now.Second  % 1 == 0 && now.Second == 1;
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۸/۰۸ ۱:۵۴

برای این حالت در متد RunAt فقط بنویسید return true. چون این متد 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

دقیقا 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 خطا 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 پروپرتیس شدم و استارترشو عوض کردم درست شد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۰۳ ۱۰:۵۵

- این پروژه باید یکبار کامل کامپایل شود و همچنین پوشه‌ی bin آن موجود باشد.
- خطای Could not load type صرفا به معنای عدم کامپایل شدن برنامه و یا نبود فایل‌های dll مرتبط در پوشه‌ی bin است.
مراجعه کنید به منوی Build و گزینه‌ی Rebuild solution را انتخاب کنید.

نویسنده: سعید شیرعلی
تاریخ: ۱۳۹۳/۰۹/۰۳ ۱۲:۴۶

سلام؛ میخوام 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 تعریف 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 نوشتید «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

ممنون؛ 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 متد بیدارکننده! رو صدا میزنه خطاهای Server 503 Unavailable و Internal Server Error 500 پرت (throw میشه).

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۰۹ ۱۸:۱۳

- از ELMAH برای ثبت خودکار ریز جزئیات خطاهای مدیریت نشده و بررسی آتی آن‌ها استفاده کنید.
- همچنین در کتابخانه‌ی فوق به قسمت ScheduledTasksCoordinator.Current.OnUnexpectedException هم دقت داشته باشد؛ مطابق مثال ارائه شده.

عنوان: **بررسی خروجی IsAjaxRequest در درخواست های $http توسط AngularJS**
نویسنده: ناصر طاهری
تاریخ: ۱۳۹۳/۰۱/۱۲ ۲۱:۱۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, Security, AngularJS

در یکی از پروژه‌هایی که دارم از AngularJS و ASP.NET MVC استفاده میکنم. به هنگام استفاده از درخواست‌های 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 بدهیم 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 برمیگرداند. در حالیکه با متدهای ساده ایجکسی Jquery مثل $.get و... ، مقدار این اکستنشن True میشود و به خوبی هم کار میکند.
درخواست‌های هر دو مورد را که با فایرباگ بررسی کردم به این مقادیر برخوردم.
ویژگی‌های درخواست توسط Jquery - $.get

```
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=ZuW8imM5USKOyBH2kih4o
                   T38mP9854FoY2y817pXhqjOyO9eQ8VUTsln9A_ufopXaC0btSjSKb8A0auWz5hFd5qiaiVO7eZxY
                   ; captchastring=55-57-EB-27-6B-06-A7-C4-CC-93-9F-4A-C8-AF-28-63-93-9E-8E-6E-
                   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=CE5EF7C684F13C1FAA8DB8F8B43E14E47587D7E9C590EB0D492620F5E01213F7
                   0D6E99B1571F1CEC9E3B97BAF77F3FB5C95E92A7462B6490D6B5ECD4FF38F642DE82D8675536
                   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
```

و ویژگی‌های درخواست توسط سرویس AngularJs - $http :

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=ZuW8imM5USKOyBH2kih4T38mP9854FoY2y817pXhqjOyO9eQ8VUTsln9A_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-eAIOyU0RQWwAYiw-fNGa6q6MK0zCso9uQWGiP-O0aezj2yC9wV_xDYY40C0g8kEl5yqMlhmMY4hmkWU6-IlqkRNG9inON8e4ybE5Yc00uZbh-XN0GPgWPM1BvS4i2J3dcMn00af2-f4wPL4qvakegYDJCkMR2NAx8Ku7HzV2JvpKcYpyfQLF-WMo19yK6fUKJwlTJXgWaZVzs4Jkq-GO9RmRuXBAsDXJTYmJMlyWpHICBwkvj_9hQy2Bdf2jggtYTVEE7XLNPw35nuiRUlgbkT3pCVd_Ipoa77aXXz0ClA1j1C-aQWZq__IErhh-4ysOTa1TJgFP-AloFKsD6Tke3a5Q1zqvKQQIOY9SohCEI-NJSNpl61VhpJBtPK
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 متد مقدار False را برمیگرداند.

بعد از کمی جستجو در این مورد ، به مخزن git انگیولار 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 رو 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 برمیگرداند.

عنوان: **بارگذاری فایل‌های ایستا از پوشه‌ی Views در ASP.NET MVC**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۲۰ ۲۳:۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

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 آن‌را مطابق معمول در پوشه‌ی Scripts در کنار سایر فایل‌ها قرار داد و یا ... حتی شاید بهتر باشد این فایل اسکریپت ویژه را در همان پوشه‌ی 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 پوشه‌ی 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 پوشه‌ی 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 پوشه‌ی 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>
```

عنوان: **هدایت خودکار آدرس‌های یافت نشد در یک سایت ASP.NET MVC به جستجوی سایت**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۱/۲۲ ۱۲:۱۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, Dependency Injection

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 متد پیش فرضی با فرمت زیر مدیریت می‌شود:

```
[ValidateInput(false)] //برنامه نویس‌ها نیاز دارند تگ‌ها را جستجو کنند
  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 را وارد کند، controllerType 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 جاری، 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 بود، مقدار پیش فرض action همان test می‌باشد. بنابراین صرف بازگشت وهله‌ای از

SearchController تمام موارد را پوشش نمی‌دهد و نیاز است دقیقا 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 متد 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; //نیاز هست دانلود عمومی فایل‌ها تحت کنترل قرار گیرد
```

- در ادامه نیاز است مسیریابی Catch all 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;
        }
    }
```

UrlConstraint مطابق تنظیم CatchAllRoute فقط زمانی فراخوانی خواهد شد که برنامه به این مسیریابی خاص برسد (و نه در سایر حالات متداول کار با کنترلر جستجو). در اینجا فرصت خواهیم داشت تا مقدار term را به RouteValueDictionary آن اضافه کنیم.

عنوان: راه‌های متفاوت رندر لایه‌ها در ASP.NET MVC
نویسنده: محمد رضا صفری
تاریخ: ۱۳۹۳/۰۱/۲۵ ۰:۵۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, Layouts

در MVC لایه‌ها (Layouts) 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) می‌تواند شامل قالب‌های CSS مختلف، کدهای Javascript مختلف و قالب بندی‌های 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 1 : استفاده از _ViewStart موجود در ریشه‌ی 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 بدید تا سیستم از آن استفاده کند.

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

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

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

می‌توان نوشت:

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

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

نصب کتابخانه‌ی Microsoft ASP.NET MVC Futures

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

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

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

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

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

یک نکته‌ی مهم

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

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

ActionLinkهای جدید بسته‌ی Microsoft.AspNet.Mvc.Futures

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

RedirectToAction جدید بسته‌ی Microsoft.AspNet.Mvc.Futures

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

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

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

برای مثال

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

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

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

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

            return View();
        }
    }
}
```

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

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

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

برای مطالعه بیشتر

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

Using the Features of ASP.NET MVC 3 Futures

عنوان: ASP.NET MVC و Identity 2.0 : مفاهیم پایه
نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۳/۰۲/۰۸ ۱:۱۵
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, MVC, Security, ASP.NET Identity

در تاریخ 20 مارچ 2014 تیم ASP.NET نسخه نهایی Identity 2.0 را منتشر کردند . نسخه جدید برخی از ویژگی‌های درخواست شده پیشین را عرضه می‌کند و در کل قابلیت‌های احراز هویت و تعیین سطح دسترسی ارزنده ای 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. بعنوان 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 مثلا احراز هویت مبتنی بر پروتوکل 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. مثلا آدرس ایمیل و اطلاعات تماس را هم در بر می‌گیرد
احراز هویت دو مرحله ای (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 (مثلا هنگام تغییر کلمه عبور)

بهبود پشتیبانی از Social log-ins
یکپارچه سازی ساده با Claims-based Authorization

Identity 2.0 تغییرات چشم گیری نسبت به نسخه قبلی به‌وجود آورده است. به نسبت ویژگی‌های جدید، پیچیدگی‌هایی نیز معرفی شده‌اند. اگر به تازگی (مانند خودم) با نسخه 1 این فریم ورک آشنا شده و کار کرده اید، 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 بعنوان یک نقطه شروع برای آشنایی با این فریم ورک نگاه کنید.

اگر به دنبال اطلاعات بیشتر و بررسی‌های عمیق‌تر هستید، لینک هایی در انتهای این مقاله نگاشت شده اند. همچنین طی هفته‌های آینده چند مقاله تخصصی‌تر خواهم 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 گرفته اند، چطور کار می‌کنند و اجزای کلی سیستم چگونه پیکربندی می‌شوند. گرچه، اگر به برنامه نویسی دات نت (ASP.NET, C#) تسلط دارید و با نسخه قبلی Identity هم کار کرده اید، درک کدهای جدید کار ساده ای خواهد بود.

Identity 2.0 با نسخه قبلی سازگار نیست

اپلیکیشن هایی که با نسخه 1.0 این فریم ورک ساخته شده اند نمی‌توانند 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 اپلیکیشن‌ها نیز دستخوش تغییراتی شده است. مهاجرت از نسخه 1.0 به 2.0 نیاز به نوشتن کدهای جدید و اعمال تغییرات متعددی دارد که از حوصله این مقاله خارج است. فعلا همین قدر بدانید که این مهاجرت نمی‌تواند بسادگی در قالب Plug-in and play صورت پذیرد!

شروع به کار : پروژه مثال‌ها را از NuGet دریافت کنید

در حال حاظر (هنگام نوشتن این مقاله) قالب پروژه استانداردی برای اپلیکیشن‌های ASP.NET MVC که ا ز Identity 2.0 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 بسازید و قالب Empty را در دیالوگ تنظیمات انتخاب کنید.

کنسول Package Manager را باز کنید و با اجرای فرمان زیر پروژه مثال‌ها 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 پروژه‌های استاندارد MVC مشاهده خواهید کرد. پروژه شما شامل قسمت‌های Models, Views, Controllers و کامپوننت‌های دیگری برای شروع به کار است. گرچه در نگاه اول ساختار پروژه بسیار شبیه به پروژه‌های استاندارد ASP.NET MVC به نظر می‌آید، اما با نگاهی دقیق‌تر خواهید دید که تغییرات جدیدی ایجاد شده‌اند و پیچیدگی هایی نیز معرفی شده اند.

پیکربندی Identity : دیگر به سادگی نسخه قبلی نیست

به نظر من یکی از مهم‌ترین نقاط قوت فریم ورک Identity یکی از مهم‌ترین نقاط ضعفش نیز بود. سادگی نسخه 1.0 این فریم ورک کار کردن با آن را بسیار آسان می‌کرد و به سادگی می‌توانستید ساختار کلی و روند کارکردن کامپوننت‌های آن را درک کنید. اما همین سادگی به معنای محدود بودن امکانات آن نیز بود. بعنوان مثال می‌توان به تایید حساب‌های کاربری یا پشتیبانی از احراز هویت‌های 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 می‌کنیم.

در هر دو نسخه، فایلی بنام *Startup.cs* در مسیر ریشه پروژه خواهید یافت. در این فایل کلاس واحدی بنام Startup تعریف شده است که متد ()ConfigureAuth را فراخوانی می‌کند. چیزی که در این فایل مشاهده نمی‌کنیم، خود متد ConfigureAuth است. این بدین دلیل است که مابقی کد کلاس Startup در یک کلاس پاره ای (Partial) تعریف شده که در پوشه *App_Start* قرار دارد. نام فایل مورد نظر *Startup.Auth.cs* است که اگر آن را باز کنید تعاریف یک کلاس پاره ای بهمراه متد ()ConfigureAuth را خواهید یافت. در یک پروژه که از نسخه Identity 1.0 استفاده می‌کند، کد متد ()ConfigureAuth مطابق لیست زیر است.

```
public partial class Startup
{
    public void ConfigureAuth(IAppBuilder 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 کوکی‌ها را مشاهده می‌کنید. همچنین کدهایی بصورت توضیحات وجود دارد که به منظور استفاده از تامین کنندگان خارجی مانند گوگل، فیسبوک، توییتر و غیره استفاده می‌شوند. حال اگر به کد این متد در نسخه Identity 2.0 دقت کنید خواهید دید که کد بیشتری نوشته شده است.

```
public partial class Startup {
    public void ConfigureAuth(IAppBuilder app) {
        // Configure the db context, user manager and role
        // manager to use a single instance per request
```

```
        app.CreatePerOwinContext(ApplicationDbContext.Create);
        app.CreatePerOwinContext<ApplicationUserManager>(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 وهله سازی می‌کنند. این وهله‌ها سپس توسط فراخوانی متد context.Get() قابل دسترسی خواهند بود. این به ما می‌گوید که حالا Owin بخشی از اپلیکیشن ما است و فریم ورک Identity 2.0 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 جدیدی برای کوکی‌ها تعریف شده است که در نسخه قبلی 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 Identity را انجام می‌دهد. کامپوننت‌های پیکربندی و کدهای کمکی دیگری نیز وجود دارند که در کلاس *IdentityConfig.cs* تعریف شده اند.

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 عمدتا توسط منطق امنیتی این فریم ورک استفاده می‌شوند و کمک شایانی در مدیریت اطلاعات کاربران به ما نمی‌کنند.

اگر از نسخه Identity 1.0 استفاده کرده باشید و مطالعاتی هم در این زمینه داشته باشید، می‌دانید که توسعه کلاس کاربران بسیار ساده است. مثلا 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 Identity 2.0 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 تغییراتی بوجود آورده اند تا نیازهای رایج توسعه دهندگان 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 IdentityUser 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 کاملا 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)ی 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 TKey) به ما اجازه می‌دهد که نوع این فیلد را مشخص کنیم. همانطور که مشاهده می‌کنید خاصیت Id در این کلاس نوع داده TKey را باز می‌گرداند.

```
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 Identity 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 Identity 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 آبجکت هایی از نوع IdentityRole و IdentityUser 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 باشید (مثلا تعریف کلاس Role را توسعه داده باشید، یا سیستمی مانند Group-based Roles را پیاده سازی کرده باشید) این تغییرات سیستم قبلی شما را خواهد شکست.

بررسی پیاده سازی جدید IdentityUser از حوصله این مقاله خارج است. فعلا 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 Identity 2.0 و کدهای کمکی

گرچه متد ()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 Identity 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 حاظر تمام این کلاس‌ها در فضای نام ApplicationName.Models قرار دارند.

Application User Manager و Application Role Manager

اولین چیزی که در این فایل به آنها بر می‌خوریم دو کلاس ApplicationUserManager و ApplicationRoleManager 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,
        IOwinContext 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 نقش ها. این کلاس بخودی خود از کلاس UserManager<ApplicationUser> ارث بری می‌کند بنابراین تمام قابلیت‌های UserManager در این کلاس هم در دسترس است. اگر به متد Create() دقت کنید می‌بینید که وهله ای از نوع ApplicationUserManager باز می‌گرداند. بیشتر تنظیمات پیکربندی و تنظیمات پیش فرض کاربران شما در این متد اتفاق می‌افتد.

مورد حائز اهمیت بعدی در متد Create() فراخوانی context.Get<ApplicationDBContext>() است. بیاد بیاورید که پیشتر نگاهی به متد ConfigAuth() داشتیم که چند فراخوانی CreatePerOwinContext داشت که توسط آنها Callback هایی را رجیستر می‌کردیم. فراخوانی متد context.Get<ApplicationDBContext>() این Callbackها را صدا می‌زند، که در اینجا فراخوانی متد استاتیک 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 متد Create() انجام می‌شوند و سپس وهله ای از نوع خود کلاس ApplicationUserManager بازگشت داده می‌شود. همچنین سرویس‌های احراز هویت دو مرحله ای نیز در همین مرحله پیکربندی می‌شوند. اکثر پیکربندی‌ها و تنظیمات نیازی به توضیح ندارند و قابل درک هستند. اما احراز هویت دو مرحله ای نیاز به بررسی عمیق‌تری دارد. در ادامه به این قسمت خواهیم پرداخت. اما پیش از آن نگاهی به کلاس ApplicationRoleManager بیاندازیم.

```
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 ApplicationUserManager مشاهده می‌کنید که کلاس ApplicationRoleManager از RoleManager<IdentityRole> ارث بری می‌کند. بنابراین تمام قابلیت‌های کلاس پایه نیز در این کلاس در دسترس هستند. یکبار دیگر متدی بنام Create() را مشاهده می‌کنید که وهله ای از نوع خود کلاس بر می‌گرداند.

سرویس‌های ایمیل و پیامک برای احراز هویت دو مرحله ای 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 مرحله ای و تایید حساب‌های کاربری را بسازید.

```
public class EmailService : IIdentityMessageService
{
    public Task SendAsync(IdentityMessage message)
    {
        // Plug in your email service here to send an email.
        return Task.FromResult(0);
    }
}
```

```
public class SmsService : IIdentityMessageService
{
    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 IIdentityMessageService پیاده سازی می‌کنند. همچنین قطعه کد زیر را از متد 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 متد Create() کلاس‌های EmailService و SmsService وهله سازی شده و توسط خواص مرتبط به وهله ApplicationUserManager ارجاع می‌شوند.

کلاس کمکی SignIn

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 متدها می‌توانیم به کنترلر AccountController در پوشه Controllers مراجعه کنیم. اما پیش از آن بگذارید نگاهی به خود کلاس SignInHelper داشته

باشیم.

```
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 نسبتا طولانی است، و بررسی عمیق آنها از حوصله این مقاله خارج است. گرچه اگر به دقت یکبار این کلاس را مطالعه کنید می‌توانید براحتی 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. متد آشنایی بنام SignInAsync() را می‌بینیم، و متدهای دیگری که مربوط به احراز هویت دو مرحله ای 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.

بعنوان مثال متد Login در کنترلر 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 doen'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.");

            return View(model);
    }
}
```

مقایسه Sign-in با نسخه Identity 1.0

در نسخه 1.0 این فریم ورک، ورود کاربران به اپلیکیشن مانند لیست زیر انجام می‌شد. اگر متد 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 متدی در کلاس UserManager را فراخوانی می‌کنیم که مشابه قطعه کدی است که در کلاس SignInHelper دیدیم. همچنین متد SignInAsync را فراخوانی می‌کنیم که مستقیما 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);
}
```

مسلما با عرضه قابلیت‌های جدید در Identity 2.0 و تغییرات معماری این فریم ورک، پیچیدگی هایی معرفی می‌شود که حتی در امور ساده ای 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 پیاده سازی پیش فرض EF فریم ورک است، که اپلیکیشن شما توسط آن داده‌های مربوط به Identity را ذخیره و بازیابی می‌کند.

در پروژه مثال ها، تیم Identity این کلاس را بطور متفاوتی نسبت به نسخه 1.0 پیکربندی کرده اند. اگر فایل *IdentityModels.cs* را باز کنید تعاریف کلاس ApplicationDbContext را مانند لیست زیر خواهید یافت.

```
public class ApplicationDbContext : IdentityDbContext<ApplicationUser> {
    public ApplicationDbContext()
        : base("DefaultConnection", throwIfV1Schema: false) {
    }

    static ApplicationDbContext() {
        // Set the database intializer which is run once during application start
        // This seeds the database with admin user credentials and admin role
        Database.SetInitializer<ApplicationDbContext>(new ApplicationDbInitializer());
    }

    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 ApplicationDbInitializer تعریف شده باشد را اجرا می‌کند. اگر به فایل *IdentityConfig.cs* مراجعه کنیم می‌توانیم تعاریف این کلاس را مانند لیست زیر بیابیم.

```
public class ApplicationDbInitializer
    : 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 مجددا 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. بعنوان مثال هنگام توسعه اپلیکیشن و بمنظور تست می‌توانیم از این رویکرد استفاده کنیم تا هر بار با دیتابیسی (تقریبا) خالی شروع کنیم.

نکته حائز اهمیت دیگر متد InitializeIdentityForEF() است. این متد کاری مشابه متد Seed() انجام می‌دهد که هنگام استفاده از مهاجرت‌ها (Migrations) از آن استفاده می‌کنیم. در این متد می‌توانید 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 منتسب 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 Identity 2.0 در پروژه‌های ASP.NET MVC داشتیم. کامپوننت‌های مختلف فریم ورک و نحوه پیکربندی آنها را بررسی کردیم و با تغییرات و قابلیت‌های جدید به اختصار آشنا شدیم. در مقالات بعدی بررسی هایی عمیق‌تر خواهیم داشت و با نحوه استفاده و پیاده سازی قسمت‌های مختلف این فریم ورک آشنا خواهیم شد.

مطالعه بیشتر

MSDN: Announcing RTM of ASP.NET Identity 2.0.0

Identity 2.0 on Codeplex

نظرات خوانندگان

نویسنده: محمّد 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 ASP.NET Identity 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 پروژه ای وجود داره که با استفاده از identity 2.0 سیستم گروپ بیس رو پیاده سازی کرده باشه؟
نمونه identity 1.0 رو پیدا کردم ولی هرکاری 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 دیتااکسسمون ADO Entity Data Model یا همون edmx باشه.چطوری می‌تونیم از Identity استفاده کنیم؟(البته از MVC 5 و EF 6 دارم استفاده می‌کنم
و دوم اینکه جداول لازم رو خودمون باید بسازیم یا مثل Membership خودش میسازه؟

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 Library جداگانه‌ای به‌منظور Migrate کردن بانک استفاده میکنه، از Identity هم استفاده کرد؛ با وجود اینکه به Identity Code 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.
مثلا پیاده سازی اون و تفکیک تو لایه‌های سرویس و دامین و لایه نمایش واقعا ادم گیج میکنه
هر چند زیاد هم نمی‌توان الگوی واحد کار باهاش به کار برد - البته نظر شخصی منه

نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۰۹/۲۹ ۱۳:۴۲

« 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 »

نویسنده: آذرنیوا
تاریخ: ۱۳۹۳/۱۱/۲۷ ۷:۲۴

نحوه تعیین سطح دسترسی در asp form برای متدها و فرم‌ها ها چه گونه است مثل کاری که فیلتر authorize در mvc انجام می‌ده

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۱۱/۲۷ ۱۰:۱۹

افزودن ASP.NET Identity به یک پروژه Web Forms

عنوان: **ارسال PingBack در ASP.NET**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۰۸ ۷:۵
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, MVC, Regular expressions, Html Agility Pack, SEO

Pingback 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 عموما 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 معروف که با PHP تهیه شده‌است. 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 عموما به دو طریق ارائه می‌شود:
الف) در هدری به نام x-pingback و یا pingback
ب) در قسمتی از کدهای HTML صفحه به شکل

```
<link rel="pingback" href="pingback server">
```

برای مثال اگر به وبلاگ‌های MSDN دقت کنید، هدر x-pingback را می‌توانید در خروجی وب سرور آن‌ها مشاهده کنید:

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 = extractPingbackServiceUriFormHeaders(response);
                if (url != null)
                    return url;

                if (!isResponseHtml(response))
                    return null;

                using (var reader = new StreamReader(response.GetResponseStream()))
                {
                    return extractPingbackServiceUriFormPage(reader.ReadToEnd());
                }
            }
        }

        private static Uri extractPingbackServiceUriFormHeaders(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 extractPingbackServiceUriFormPage(string content)
        {
            if (string.IsNullOrWhiteSpace(content)) return null;
            var regex = new Regex(@"(?s)<link\srel=""pingback""\shref=""(.+?)""",
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);
        }
```

نحوه‌ی استخراج آدرس سرویس Pingback یک سایت را در کدهای فوق ملاحظه می‌کنید.

targetUri، آدرسی است از یک سایت دیگر که در سایت ما درج شده‌است. زمانیکه این صفحه را درخواست می‌کنیم، response.Headers.AllKeys 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 متد getValidAbsoluteUri، بررسی بر روی UriKind.Absolute انجام شده‌است.

اگر هدر فاقد کلید x-pingback باشد، قسمت ب را باید بررسی کرد. یعنی نیاز است محتوای Html صفحه را برای یافتن link rel=pingback بررسی کنیم. همچنین باید دقت داشت که پیش از اینکار نیاز است حتما بررسی isResponseHtml صورت گیرد. برای مثال در سایت شما لینکی به یک فایل 2 گیگابایتی SQL Server درج شده‌است. 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 ایی 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.WriteElementString("string", Uri.EscapeUriString(_sourceUri.ToString()));
                writer.WriteEndElement();
                writer.WriteEndElement();

                writer.WriteStartElement("param");
                writer.WriteStartElement("value");
                writer.WriteElementString("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 کتابخانه‌ی Cook Computing's XML-RPC.Net استفاده کرد و یا مطابق کدهای فوق، دستورات آن‌را توسط یک XmlTextWriter کنار هم قرار داد و نهایتا در درخواست Post ارسالی درج کرد.

در اینجا sourceUri آدرس صفحه‌ای در سایت ما است که targetUri ایی (آدرسی از سایت دیگر) در آن درج شده‌است. در یک pinback، صرفا این دو آدرس به سرویس دریافت کننده‌ی pingback ارسال می‌شوند.

سپس سایت دریافت کننده‌ی ping، ابتدا sourceUri را دریافت می‌کند تا عنوان آن‌را 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

نحوه‌ی استفاده از کلاس Pingback تهیه شده

کار ارسال Pingback عموما به این نحو است: هر زمانیکه مطلبی یا یکی از نظرات آن، ثبت یا ویرایش می‌شوند، نیاز است Pingbackهای آن ارسال شوند. بنابراین تنها کاری که باید انجام شود، استخراج لینک‌های خارجی یک صفحه و سپس فراخوانی متد Send کلاس فوق است.

یافتن لینک‌های یک محتوا را نیز می‌توان مانند متد extractPingbackServiceUriFormPage فوق، توسط یک Regex انجام داد و یا حتی با استفاده از کتابخانه‌ی معروف 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));
```

عنوان: **Identity 2.0 : تایید حساب های کاربری و احراز هویت دو مرحله ای**
نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۳/۰۲/۱۰ ۱:۳۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, Security, ASP.NET Identity

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 مرحله ای (Two-Factor Authorization) بود. در این پست راه اندازی این دو قابلیت را بررسی می‌کنیم.

تیم ASP.NET Identity پروژه نمونه ای 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 پروژه ای با ساختار متداول پروژه‌های ASP.NET MVC داشته باشید. به تصویر زیر دقت کنید.

Solution Explorer
Search Solution Explorer (Ctrl+;)
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
Solution Explorer
Team Explorer

همانطور که می‌بینید ساختار پروژه بسیار مشابه پروژه‌های معمول MVC است، اما آیتم‌های جدیدی نیز وجود دارند. فعلا تمرکز اصلی ما روی فایل *IdentityConfig.cs* است که در پوشه *App_Start* قرار دارد.

اگر فایل مذکور را باز کنید و کمی اسکرول کنید تعاریف دو کلاس سرویس را مشاهده می‌کنید: EmailService و SmsService.

```
public class EmailService : IIdentityMessageService
{
    public Task SendAsync(IdentityMessage message)
    {
        // Plug in your email service here to send an email.
        return Task.FromResult(0);
    }
}

public class SmsService : IIdentityMessageService
{
    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 IIdentityMessageService را پیاده سازی می‌کنند. می‌توانید از این قرارداد برای پیاده سازی سرویس‌های 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 می‌کند.

```
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 می‌بینید این قطعه کد ابتدا نقشی بنام Admin می‌سازد. سپس حساب کاربری ای، با اطلاعاتی پیش فرض ایجاد شده و بدین نقش منتسب می‌گردد. اطلاعات کاربر را به دلخواه تغییر دهید و ترجیحا از یک آدرس ایمیل زنده برای آن استفاده کنید.

تایید حساب‌های کاربری : چگونه کار می‌کند

بدون شک با تایید حساب‌های کاربری توسط ایمیل آشنا هستید. حساب کاربری ای ایجاد می‌کنید و ایمیلی به آدرس شما ارسال می‌شود که حاوی لینک فعالسازی است. با کلیک کردن این لینک حساب کاربری شما تایید شده و می‌توانید به سایت وارد شوید.

اگر به کنترلر AccountController در این پروژه نمونه مراجعه کنید متد Register را مانند لیست زیر می‌یابید.

```
[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 فراخوانی متد UserManager.SendEmailAsync را می‌یابید که آرگومانهایی را به آن پاس می‌دهیم. در کنترلر جاری، آبجکت UserManager یک خاصیت (Property) است که وهله ای از نوع ApplicationUserManager را باز می‌گرداند. اگر به فایل *IdentityConfig.cs* مراجعه کنید تعاریف این کلاس را خواهید یافت. در این کلاس، متد استاتیک ()Create وهله ای از ApplicationUserManager را می‌سازد که در همین مرحله سرویس‌های پیام رسانی پیکربندی می‌شوند.

```
public static ApplicationUserManager Create(
    IdentityFactoryOptions<ApplicationUserManager> options,
    IOwinContext 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 کلاس‌های EmailService و SmsService روی وهله ApplicationUserManager تنظیم می‌شوند.

```
manager.EmailService = new EmailService();
manager.SmsService = new SmsService();
```

درست در بالای این کدها می‌بینید که چگونه تامین کنندگان احراز هویت دو مرحله ای (مبتنی بر ایمیل و پیامک) رجیستر می‌شوند.

```
// 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 حساب‌های کاربری توسط ایمیل و احراز هویت دو مرحله ای توسط ایمیل و/یا پیامک نیاز به پیاده سازی هایی معتبر از قرارداد IIdentityMessageService دارند.

پیاده سازی سرویس ایمیل توسط ایمیل خودتان

پیاده سازی سرویس ایمیل نسبتا کار ساده ای است. برای ارسال ایمیل‌ها می‌توانید 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 : IIdentityMessageService
{
    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;

        // Creatte 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 مطمئنا می‌توانید مستندات لازم را پیدا کنید. اما در کل رویکرد مشابهی خواهید داشت.

پیاده سازی سرویس ایمیل با استفاده از SendGrid

سرویس‌های ایمیل متعددی وجود دارند اما یکی از گزینه‌های محبوب در جامعه دات نت SendGrid است. این سرویس API قدرتمندی برای زبان‌های برنامه نویسی مختلف فراهم کرده است. همچنین یک Web API مبتنی بر HTTP نیز در دسترس است. قابلیت دیگر اینکه این سرویس مستقیما با Windows Azure یکپارچه می‌شود.

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 ایمیلی 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 می‌توانید از هر آدرسی بعنوان آدرس فرستنده استفاده کنید.

```
public class EmailService : IIdentityMessageService
{
    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;

        // Creatte 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 می‌توانیم سرویس ایمیل را تست کنیم.

آزمایش تایید حساب‌های 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 بدرستی کار کند باید به صفحه ای مانند تصویر زیر هدایت شوید.

همانطور که مشاهده می‌کنید پاراگرافی در این صفحه وجود دارد که شامل لینک فعالسازی است. این لینک صرفا جهت تسهیل کار توسعه دهندگان درج می‌شود و هنگام توزیع اپلیکیشن باید آن را حذف کنید. در ادامه به این قسمت باز می‌گردیم. 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 C# API قدرتمند ارائه می‌کند. می‌توانید 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 می‌شود. شماره پیامکی خود را می‌توانید پس از ورود به سایت و پیمایش به صفحه Numbers مشاهده کنید.

شناسه‌های SID و Auth Token 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 : IIdentityMessageService
{
    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 سرویس‌های ایمیل و پیامک را در اختیار داریم می‌توانیم 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 می‌تواند 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 می‌توانید پروسه را تایید کنید. اگر همه چیز بدرستی 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 مجددا سعی کنید به سایت وارد شوید. در این مرحله یک

انتخاب به شما داده می‌شود. می‌توانید کد احراز هویت دو مرحله ای خود را توسط ایمیل یا پیامک دریافت کنید.

پس از اینکه گزینه خود را انتخاب کردید، کد احراز هویت دو مرحله ای برای شما ارسال می‌شود که توسط آن می‌توانید پروسه ورود به سایت را تکمیل کنید.

حذف میانبرهای آزمایشی

همانطور که گفته شد پروژه نمونه شامل میانبرهایی برای تسهیل کار توسعه دهندگان است. در واقع اصلا نیازی به پیاده سازی سرویس‌های ایمیل و پیامک ندارید و می‌توانید با استفاده از این میانبرها حساب‌های کاربری را تایید کنید و کدهای احراز هویت دو مرحله ای را نیز مشاهده کنید. اما قطعا این میانبرها پیش از توزیع اپلیکیشن باید حذف شوند.

بدین منظور باید نماها و کدهای مربوطه را ویرایش کنیم تا اینگونه اطلاعات به کلاینت ارسال نشوند. اگر کنترلر AccountController را باز کنید و به متد Register() بروید با کد زیر مواجه خواهید 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 می‌بینید پیش از بازگشت از این متد، متغیر callbackUrl به ViewBag اضافه می‌شود. 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.HasBeenVerified())
    {
        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 Status به ViewBag 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، شناسه‌های 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 کنترل ارسال می‌کند (مثلا مخازن عمومی مثل GitHub). بدین منظور می‌توانید یکی از پست‌های اخیر را مطالعه کنید.

نظرات خوانندگان

نویسنده: دات نت کدینگ
تاریخ: ۱۳۹۳/۰۲/۱۱ ۱۳:۱

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 ۱۳:۲۶

از ef-migrations برای اعمال تغییرات انجام شده استفاده کردید؟ بدون اینکار تغییری اعمال نمیشه.

نویسنده: دات نت کدینگ
تاریخ: ۱۳۹۳/۰۲/۱۴ ۰:۳۳

منظورتون اجرا کردن migration باشه، بله،
اینکار رو انجام دادین؟ شدنیه؟

نویسنده: علی
تاریخ: ۱۳۹۳/۰۳/۰۵ ۱۴:۱۹

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 لاگین کنه. باید کجا این مسئله چک شود و چگونه؟ ممنون

نویسنده: Sam
تاریخ: ۱۳۹۳/۰۵/۰۴ ۲۳:۳۲

من از مدل لایه ای که آقای نصیری در مباحث ان تی تی کد فرست گفتن استفاد می‌کنم ، می‌خواستم بدونم که چطور میشه کلاس‌ها user identity رو به sql server به وسیله code first و migraiton انتقال داد البته با نام‌های دلخواه جداول در sql...
ممنون میشم اگر توضیح بدید،

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۰۵ ۱۰:۲

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 مسایل و نکات آن مانند سایر مباحث متداول EF Code first است و تفاوتی نمی‌کند.

نویسنده: سام میرزاقراچه
تاریخ: ۱۳۹۳/۰۶/۰۴ ۲۲:۳

با سلام مجدد
چگونه می‌توان در Role Manager یک Property جدید اضافه کرد و یا همان بخش Custom کردن Role در Identity 2 .....

نویسنده: ربال
تاریخ: ۱۳۹۳/۰۸/۱۸ ۰:۵۰

با سلام.آیا استفاد ه از Twilio محدودیت دارد؟ یعنی اگر تو پروژه استفاده کنیم بعدا ب مشکل بر نخواهیم خورد (از جانب رایگان بودنش)
ممنون .

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۲۹ ۱۳:۴۲

« اعمال تزریق وابستگی‌ها به مثال رسمی ASP.NET Identity »

نویسنده: عباس معینی
تاریخ: ۱۳۹۴/۰۱/۱۵ ۱۲:۱

با سلام؛ من یک پروژه همانطور که در ابتدا توضیح دادید ایجاد کردم. وقتی کاربر جدیدی تعریف میکنم بدون اینکه روی لینک فعال سازی کلیک کنم، لاگین میشه! ممنون میشم راهنماییم کنید.

عنوان: **مدیریت درخواست‌های شرطی در ASP.NET MVC**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۱۲ ۰:۲۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

فرض کنید کنید هدرهای کش کردن عناصر پویا و یا ثابت سایت را برای مدتی مشخص تنظیم کرده‌اید.
سؤال: مرورگر چه زمانی از کش محلی خودش استفاده خواهد کرد (بدون ارسال درخواستی به سرور) و چه زمانی مجددا 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. دراینجا، زمانی یک درخواست غیرشرطی ارسال می‌شود که نسخه‌ی ذخیره شده‌ی محلی منبع مورد نظر، مهیا نباشد. در این حالت، اگر منبع درخواستی در سرور موجود باشد، در پاسخ ارسالی خود وضعیت 200 یا HTTP/200 OK 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-Modified-Since و یا If-None-Match است و هدف از آن سؤال پرسیدن از وب سرور است که آیا این منبع خاص، در سمت سرور اخیرا تغییر کرده‌است یا خیر؟ اگر پاسخ سرور خیر باشد، باز هم از همان کش محلی استفاده خواهد شد و مجددا 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 HTTP/304 Not Modified مشخص می‌گردد (این پاسخ هیچ body خاصی نداشته و فقط یک سری هدر است). اما اگر منبع درخواستی اخیرا تغییر کرده باشد، پاسخ HTTP/200 OK را در هدر بازگشت داده شده، به مرورگر بازخواهد گرداند (یعنی محتوا را مجددا 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، مجددا 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 گرفته‌است (برای مثال بر اساس id یا نام یک فایل).
ب) اگر نحوه‌ی هدایت به صفحه‌ی جاری 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.
ه) 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 متد، الزاما به معنای کش شدن آن تا مدت زمان 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، حتما 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 متد 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 OutputCache 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 متد GetImage تهیه می‌کند:

```
<img src="@Url.Action("GetImage","Home", new { name = "test.png"})"/>
```

در سطر اول متد GetImage، یک break point قرار دهید و سپس برنامه را توسط VS.NET اجرا کنید.

بار اول که صفحه‌ی اول برنامه درخواست می‌شود، یک چنین هدرهایی رد و بدل خواهند شد (توسط ابزارهای توکار مرورگر وب کروم تهیه شده‌است؛ همان دکمه‌ی 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 Cache-Control، Expires و Last-Modified را اضافه کرده‌است.

در همین حال اگر صفحه را ریفرش کنیم (فشردن دکمه‌ی 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 اصلا متد کش شده‌ی GetImage دیگر اجرا نخواهد شد و به break point داخل آن نخواهیم رسید)، 304 یا تغییر نکرده‌است. بنابراین مرورگر مجددا 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) مجددا 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 شده‌است from cache. یعنی درخواستی را به سرور برای دریافت فایل ارسال نکرده است. عدم رسیدن به break point داخل متد GetImage نیز مؤید آن است.

مشکل! مرورگر را ببندید، تا کار دیباگ برنامه خاتمه یابد. مجددا برنامه را اجرا کنید. مشاهده خواهید کرد که ... اجرای برنامه در Break point قرار گرفته در سطر اول متد GetImage متوقف می‌شود. چرا؟! مگر قرار نبود تا یک ماه دیگر کش شود؟! هدر رد و بدل شده نیز Status Code:200 OK کامل 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 متدی که حاصل آن نیاز است کش شود، خودمان و به صورت دستی مدیریت کنیم (فیلتر OutputCache این‌کار را انجام نمی‌دهد). 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 کاملا بسته و مجددا 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 کاهش 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.

تبدیل این 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 متدی که OutputCache دارد، تکرار شود. روش AOP وار آن در ASP.NET MVC، تبدیل این کدها به یک فیلتر با قابلیت استفاده‌ی 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.RequestContext.HttpContext.Response;
            var request = filterContext.RequestContext.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 متد، 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 OutputCache در ASP.NET MVC، هدر If-Modified-Since 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 وهله‌ی 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

عنوان: استفاده از Froala WYSIWYG Editor در ASP.NET
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۱۸ ۱۵:۰
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, MVC

چندی قبل، معرفی ادیتور سبک وزن و مناسبی را تحت عنوان RedActor ، در این سایت ملاحظه کردید. زمانیکه این‌کار 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 MVC و همچنین ASP.NET Web forms.

دریافت آخرین نگارش Froala WYSIWYG Editor

برای دریافت فایل‌های آخرین نگارش این ادیتور وب می‌توانید به سایت آن، قسمت دریافت فایل‌ها مراجعه نمائید.

http://editor.froala.com/download

و یا به این آدرس مراجعه کنید:

https://github.com/froala/wysiwyg-editor/releases

ساختار پروژه و نحوه‌ی کپی فایل‌های آن

در هر دو مثالی که فایل‌های آن‌را از انتهای بحث می‌توانید دریافت کنید، این ساختار رعایت شده است:

فایل‌های CSS و فونت‌های آن، در پوشه‌ی Content قرار گرفته‌اند.

فایل‌های اسکریپت و زبان آن (که دارای زبان فارسی هم هست) در پوشه‌ی Scripts کپی شده‌اند.

یک نکته

فایل font-awesome.css را نیاز است کمی اصلاح کنید. مسیر پوشه‌ی فونت‌های آن اکنون با fonts شروع می‌شود.

تنظیمات اولیه

تفاوتی نمی‌کند که از وب فرم‌ها استفاده می‌کنید یا MVC، نحوه‌ی تعریف و افزودن پیش نیازهای 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 master یا layout سایت اضافه شوند.

استفاده از Froala WYSIWYG Editor در ASP.NET MVC

در ادامه نحوه‌ی فعال سازی ادیتور وب Froala را در یک View برنامه‌های ASP.NET 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>
}
```

اگر می‌خواهید فونت پیش فرض آن را تنظیم کنید، باید مطابق کدهای ابتدای فایل، ویژگی‌های froala-element را تغییر دهید.
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 language و direction ملاحظه می‌کنید.

پارامترهای saveURL، autosave 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 این‌کار باید انجام شود.

```
/// <summary>
        /// ذخیره سازی خودکار
        /// </summary>
        [HttpPost]
        [ValidateInput(false)]
        public ActionResult FroalaAutoSave(string body, int? postId) // نام پارامتر بادی را تغییر ندهید
        {
            //todo: save body ...
            return new EmptyResult();
        }
```

در قسمت سمت سرور هم می‌توان این مقادیر ارسالی را در اکشن متدی که ملاحظه می‌کنید، دریافت کرد.
چون قرار است تگ‌های HTML به سرور ارسال شوند، ویژگی ValidateInput به false تنظیم شده‌است.
saveParams آن، برای مقدار دهی پارامترهای اضافی است که نیاز می‌باشند تا به سرور ارسال شوند. مثلا شماره مطلب جاری نیز به سرور ارسال گردد.
در اینجا نام پارامتری که ارسال می‌گردد، دقیقا مساوی body است. بنابراین آن‌را تغییر ندهید.

پارامترهای imageUploadURL و imageParams برای فعال سازی ذخیره تصاویر آن در سرور کاربرد دارند.
اکشن متد مدیریت کننده‌ی 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) // نام پارامتر فایل را
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، دقیقا معادل file است. بنابراین آن‌را تغییر ندهید.
خروجی آن برای مشخص سازی محل ذخیره سازی تصویر در سرور باید یک خروجی JSON دارای خاصیت و پارامتر link به نحو فوق باشد (این مسیر، یک مسیر نسبی است؛ نسبت به ریشه سایت).
imageParams آن برای مقدار دهی پارامترهای اضافی است که نیاز می‌باشند تا به سرور ارسال شوند. مثلا شماره مطلب جاری نیز به سرور ارسال گردد.

استفاده از Froala WYSIWYG Editor در ASP.NET Web forms

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 requestValidationMode به نگارش 2 تنظیم گردیده‌است تا بتوان توسط این ادیتور تگ‌های ارسالی را به سرور ارسال کرد. به علاوه ClientIDMode=Static نیز تنظیم شده‌است، تا بتوان از ID تکست باکس قرار گرفته در صفحه، به سادگی در کدهای سمت کاربر جی‌کوئری استفاده کرد.

اگر دقت کرده باشید، save urlها اینبار به فایل FroalaHandler.ashx اشاره می‌کنند. 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 وب ای پی آی استفاده شود
        //todo: یا دو هندلر 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;

                    //todo: در اینجا مسایل امنیتی آپلود 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 نحوه‌ی مدیریت سمت سرور auto save و همچنین ارسال تصاویر ادیتور Froala ، ذکر شده‌اند. با استفاده از context.Request.Form می‌توان به عناصر ارسالی به سرور دسترسی پیدا کرد. همچنین توسط context.Request.Files، اگر فایلی ارسال شده بود، ذخیره شده و نهایتا خروجی JSON مدنظر بازگشت داده می‌شود.

یک 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، حتما مدنظر داشته باشید. در اینجا فرض شده‌است که پوشه‌ی uploads قرار است قابلیت اجرای فایل‌های پویا را نداشته باشد.

کدهای کامل این مطلب را در ادامه می‌توانید دریافت کنید

Froala-Sample

نظرات خوانندگان

نویسنده: علی رضایی
تاریخ: ۱۳۹۳/۰۲/۱۸ ۱۸:۱۱

سلام
ممنون برای این مطلب
فکر کنم این ادیتور با کروم یک مشکل داره:
برای مثال کلمه «تست» رو در اون تایپ کنید، سپس بعد از «س» یک حرف جدید مثل «ن» وارد کنید: نتیجه باید بشود «تسنت» اما میشود «تسنت». البته من در کروم 34 تست 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 مهربانی
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 آن‌را 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 نوشتیم، 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><o
ption>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();
                            });
                        }
                    }
                }
```

نویسنده: احسان میرسعیدی
تاریخ: ۱۳۹۳/۰۳/۱۱ ۱:۱۳

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 SyntaxHighliter 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 گزاری دانش تون

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۳/۰۴/۰۱ ۱۱:۲۷

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 ۱۳:۴۳

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 ۱۳:۴۶

این قابلیت را به صورت توکار دارد. اطلاعات بیشتر

نویسنده: حمید رضا رحیمی
تاریخ: ۱۳۹۳/۰۴/۲۴ ۱۴:۱۲

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 نشده Enables or disables adding the rel="nofollow" attribute on all links.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۲۴ ۱۴:۱۷

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 imagesLoadURL در 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 آن‌را به فرمت JSON بازگشت دهد.

نویسنده: رضایی
تاریخ: ۱۳۹۳/۰۵/۲۷ ۲۲:۵۵

برای حذف یک تصویر از قسمت منیجر از خاصیت imageDeleteURL و برای ارسال پارامترها از imageDeleteParams استفاده می‌کنیم. خاصیت imageDeleteParams را باید به چه صورت مقداردهی کرد
موقعی که از کد زیر استفاده میکنم ادیتور لود نمیشه

```
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. 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

ممنون؛ پارامتر src که آدرس تصویر رو در خودش داره به سمت سرور ارسال میشه.

نویسنده: رضایی
تاریخ: ۱۳۹۳/۰۵/۲۸ ۲۲:۱۲

با سلام

موقعی که از ادیتور توی یک modal استفاده میکنم تکست باکس‌های ادیتور 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

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۲۸ ۲۲:۵۷

اینجا گزارش شده . در حال بررسی است.

نویسنده: Mohammadreza AAA
تاریخ: ۱۳۹۳/۰۶/۰۲ ۱۲:۵

سلام.
من این ادیتورو گذاشتم. فوق العاده راحت و قشنگه. فقط با بخش ارسال عکسش مشکل دارم.
مسئله اینکه عکس من توی پوشه مورد نظرم داره ذخیره میشه ولی اشکالش اینکه توی متنم موقع نمایش هیچی نمیاید. همه خصوصیات مثلا ایتالیک و بولد به متون اعمال میشن ولی عکس نیست!

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۶/۰۲ ۱۲:۱۱

بررسی کنید چه خطایی در فایرباگ گزارش شده

نویسنده: M.Shakeri
تاریخ: ۱۳۹۳/۰۶/۰۴ ۸:۵۷

سلام
اگه داخل ادیتور متنی نوشته نشه مقدار پیش فرضش

```
"<p><br></p>"
```

هست که موقع اعتبار سنجی معتبره.
تو تنظیماتش هم گزینه ای برای خالی کردن ادیتور وجود نداره.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۶/۰۵ ۱۲:۵۰

« اعتبار سنجی سمت کاربر wysiwyg-editorها در ASP.NET MVC »

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۶/۱۷ ۱۳:۲۴

امکان آپلود فایل هم به آن اضافه شده: ^ و ^

نویسنده: هادی دانش پژوه
تاریخ: ۱۳۹۳/۰۶/۱۸ ۹:۴۳

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: ۱۳۹۳/۰۶/۱۸ ۱۰:۱۶

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: ۱۳۹۳/۰۷/۰۸ ۱۷:۱۹

سلام؛ چطوری می‌توان به این ویرایشگر قابلیت Code اضافه کرد آیا شما قبلا تجربه همچین کاری را دارید و یا حتی جایی دیده‌اید؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۰۸ ۱۷:۲۹

کمی بالاتر توضیح دادم.

نویسنده: اصغر
تاریخ: ۱۳۹۳/۰۷/۲۵ ۱۱:۱۹

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 نکته ای داره ممنون میشم بهم بگید. تو اینترنت هم خیلی گشتم ولی چیز خاصی پیدا نکردم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۲۵ ۱۱:۴۳

مشکل CSS دارید. سطر:

```
var htmlCode = "<pre language='" + lang + "' name='code'>" + code + "</pre></div>";
```

مطابق تنظیمات شخصی است. اینجا class لازم را بر اساس تنظیمات CSS نمایشگر کدهای خود اضافه کنید (مثلا class='brush:' csharp).

```
var htmlCode = "<pre class='brush: " + lang + "' language='" + lang + "' name='code'>" + code +
"</pre></div>";
```

نویسنده: اصغر
تاریخ: ۱۳۹۳/۰۷/۲۵ ۱۲:۱۱

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 همش توی یک خط نمایش داده میشه و اسکرول افقی میخوره. مثل شکل زیر:

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۲۵ ۱۲:۱۸

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: ۱۳۹۳/۱۰/۱۴ ۱۹:۲۳

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: ۱۳۹۳/۱۰/۱۴ ۲۱:۵۱

در مثالی که پیوست شد، از فایل jquery.unobtrusive-ajax 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).

نویسنده: رضا منصوری
تاریخ: ۱۳۹۳/۱۱/۲۳ ۱۶:۰

بر اساس این باید از this به جای editor استفاده کنیم.
من به این صورت استفاده کردم

```
insertHTML: {
                                title: 'Insert Code',
                                icon: {
                                    type: 'font',
                                    value: 'fa fa-dollar' // Font Awesome icon class fa fa-*
                                },
                                callback: function (editor) {
                                    this.saveSelection();
                                    var thisEditor = this;
                                    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;' /><label>Language:</label><select
id='code_lang'><option>CSharp</option><option>VB</option><option>JScript</option><option>Sql</option><o
ption>XML</option><option>CSS</option><option>Java</option><option>Delphi</option></select> <input
type='button' name='insert' id='insert_btn' value='Insert' />";
                                    $(dialog).appendTo(wrapper);

                                    $("#code_area").text(this.text());

                                    if (!this.selectionInEditor()) {
                                        this.$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 class='brush: " + lang.toLowerCase() + "'
language='" + lang + "' name='code'>" + code + "</pre></div>"; // syntaxhighlighter با این کد هماهنگ
است
                                        //var htmlCode = "<pre language='" + lang + "' name='code'>" +
code + "</pre></div>";
                                        var codeBlock = "<div align='left' dir='ltr'>" + htmlCode +
"</div>";

                                        thisEditor.restoreSelection();
                                        thisEditor.insertHTML(codeBlock);
                                        thisEditor.saveUndoStep();

                                        codeModal.remove();
                                    });
                                }
                            }
```

نویسنده: سانی
تاریخ: ۱۳۹۳/۱۲/۲۵ ۱۶:۳۰

اما این کدی که به عنوان custombutton گذاشتین رو توی کدوم فایل و به چه صورت و به کجای فایلی که معرفی می‌فرمایید باید وارد کرد؟
، اجازه بفرمایید تصحیح کنم ،
این کد را مطابق دستور شما اضافه کردم اما زمانی کلیک میکنم هیچ اتفاقی نمیافته

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۲/۲۵ ۱۶:۳۸

- سورس انتهای بحث را دریافت کنید. به خاصیت buttons آن (که در View برنامه موجود است)، مقدار insertHTM اضافه می‌شود. همچنین یک خاصیت جدید هم به نام customButtons با مقداری که مشخص شده، باید در کنار سایر خواص این افزونه، اضافه شود.
- سایر نظرات تکمیلی و نگارش‌های بهبود یافته را هم مدنظر داشته باشید.

نویسنده: سانی
تاریخ: ۱۳۹۳/۱۲/۲۵ ۱۷:۵۳

از راهنمایی شما بسیار متشکرم. دو مورد برای من پیش اومد؛ یکی اینکه با استفاده از راهنمایی‌های شما دکمه کد اضافه شد، اما اول اینکه زمانی روی آن کلیک میکنم هیچ اتفاقی نمیافته. با دو browser چک کردم و هیچ تغییری نکرد. مطلب دوم: در local آیکن‌ها نمایش داده می‌شوند اما روی هاست آیکن‌ها نمایش داده نمی‌شوند. در ابتدا از bundle استفاده می‌کردم اما با اینکه لینک مستقیم فایل‌ها را هم گذاشتم باز همان مشکل را داشتم. البته فایل اصلی jquery را هم به ابتدای صفحه انتقال دادم، اما باز هم تغییری نکرد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۲/۲۵ ۲۳:۱۴

کدهای این مطلب به مخزن کد Froala-Sample منتقل شدند.
- نحوه‌ی تعریف دکمه‌ی کد در اینجا
- آیکن‌ها از پوشه‌ی fonts قرار گرفته در ریشه‌ی سایت خوانده می‌شوند. اگر سرور قادر به نمایش آن‌ها نیست به احتمال زیاد mime type آن‌ها را تعریف نکرده‌است (^ و ^ و ^).

```
<system.webServer>
  ...
  <staticContent>
    <remove fileExtension=".woff" />
    <mimeMap fileExtension=".woff" mimeType="application/font-woff" />
    <mimeMap fileExtension=".eot" mimeType="application/vnd.ms-fontobject" />
    <mimeMap fileExtension=".otf" mimeType="font/otf" />
    <mimeMap fileExtension=".svg" mimeType="images/svg+xml" />
    <mimeMap fileExtension=".svgz" mimeType="images/svg+xml" />
  </staticContent>
</system.webServer>
```

نویسنده: احمد نواصری
تاریخ: ۱۳۹۴/۰۱/۱۳ ۰:۳۷

سلام. من در حال نوشتن یک سیستم CMS هستم و برای نوشتن مطلب از این ادیتور استفاده میکنم. وقتی مطلب رو ارسال میکنم و میخوام در قسمت مطالب نشون بدم، محتوای نوشته به صورت کد HTML نشون داده میشه، میخواستم بدونم چطور میتونم این مشکل رو برطرف کنم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۴/۰۱/۱۳ ۱:۴۱

باید از متد @Html.Raw برای جلوگیری از encoding خودکار razor استفاده کنید.

نویسنده: احمد نواصری
تاریخ: ۱۳۹۴/۰۱/۱۳ ۱۱:۲۰

آیا استفاده از این متد، مشکلی از نظر امنیتی ایجاد نخواهد کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۴/۰۱/۱۳ ۱۱:۴۲

ورودی کاربر را همیشه باید پردازش و تمیز کنید و یا از هدرهای Content Security Policy استفاده کنید.

عنوان: استفاده از نگارش سوم Google Analytics API در سرویس‌های ویندوز یا برنامه‌های وب
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۲۴ ۷:۵
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, MVC, google analytics

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 عملا در یک برنامه‌ی وب یا سرویس ویندوز قابل اجرا نخواهند بود. زیرا در حالت متداول 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 تائید شده‌ی اکانت در حال استفاده‌ی از API، در پوشه‌ی AppData ویندوز برای استفاده‌های بعدی ذخیره شود. این مورد برای یک برنامه‌ی دسکتاپ 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 عموما 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 X509 Certificate 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 آن‌را 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 نمائید 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 نمائید.

در صفحه‌ی باز شده، گزینه‌ی Service account را انتخاب کنید. اگر سایر گزینه‌ها را انتخاب نمائید، 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.

در اینجا ابتدا یک فایل مجوز p12 را به صورت خودکار دریافت خواهید کرد و همچنین پس از ایجاد client id، نیاز است، ایمیل آن‌را جایی یادداشت نمائید:

از این ایمیل و همچنین فایل p12 ارائه شده، جهت لاگین به سرور استفاده خواهد شد.

همچنین نیاز است تا به برگه‌ی APIs پروژه‌ی ایجاد شده رجوع کرد و گزینه‌ی Analytics API آن‌را فعال نمود:

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:

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:

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 پیشنیازهای 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 KeyFilePassword { set; get; }

        public AnalyticsAuthentication()
        {
            KeyFilePassword = "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 p12 ایی که از گوگل دریافت شد، یک X509Certificate2 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.

مثالی از نحوه استفاده از کلاس GoogleAnalyticsApiV3

در ادامه یک برنامه‌ی کنسول را 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 p12 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 شده‌است.

توسط AnalyticsQueryParameters می‌توان نحوه‌ی کوئری گرفتن از Google Analytics را مشخص کرد. تاریخ شروع و پایان گزارش گیری در آن مشخص هستند. در مورد پارامترهایی مانند Dimensions و Metrics بهتر است به مرجع کامل آن در گوگل مراجعه نمائید:

Dimensions & Metrics Reference

برای نمونه در مثال فوق، تعداد کاربران، سشن‌های 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 متد getProfile نوشته شده تفاوتی نمی‌کند. چون تمام اکانت‌های موجود را جهت یافتن آدرس سایت مدنظر جستجو می‌کند تا شماره پروفایل آن‌را برای کوئری گرفتن استخراج کند.
+ اکانت Client ID 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 گوگلی را برای این‌کار 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 p12 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 فعال است؟ اگر خیر، crypto susbsystem 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: ۱۳۹۳/۰۲/۲۴ ۱۴:۲

حالت سوم پاسخ شما برای مورد من درست کار کرد. تشکر فراوان.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۲۹ ۱۵:۲۸

یک نکته‌ی تکمیلی

اگر کتابخانه‌ی Google.Apis.Analytics.v3 را بر روی یک سیستم دات نت 4 اجرا کنید، احتمالا خطای ذیل را دریافت خواهید کرد:

```
Could not load type 'System.Net.HttpStatusCode' from assembly System.Net
```

علت اینجا است که دات نت 4 نیاز به وصله‌ی KB2468871 دارد تا بتواند portable libraries را بارگذاری کند.

نویسنده: کامران
تاریخ: ۱۳۹۳/۰۵/۲۱ ۲:۵۲

از این ایراد میگیره میگه null هست، من اکانت ساختم پراپرتی 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: ۱۳۹۳/۰۵/۲۱ ۱۰:۱۱

«من اکانت ساختم پراپرتی هم درست کردم»
کافی نیست. تمام مراحل باید انجام شوند.
این خطاها صرفا به معنای غیرمعتبر بودن یکی از مراحل است که سبب شده‌اند اطلاعات آن مرحله خاص قابل دریافت نباشد.

نویسنده: کامران
تاریخ: ۱۳۹۳/۰۵/۲۱ ۱۰:۵۳

نه اینکه فقط اکانت و پراپرتی درست کرده باشم، همه مراحل رو انجام دادم اما از خطی که خطا میده متوجه شدم که پرپرتی‌های من رو پیدا نمیکنه یعنی خالی برمیگردونه، قبلا هم کار کرده بودم همین مشکل پیش میومد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۲۱ ۱۱:۹

در متد getProfile :
- اگر account نال هست یعنی SiteUrl مشخص شده در ابتدای برنامه، در اکانت آنالیتیکس شما موجود نیست. روی سطر accountList.Items یک break-point قرار دهید تا لیست واقعی اکانت‌های ثبت شده را مشاهده کنید.
- اگر به مرحله‌ی اجرای accountList نمی‌رسید، یعنی ServiceAccountEmail معتبری وارد نشده‌است.

نویسنده: کامران
تاریخ: ۱۳۹۳/۰۵/۲۱ ۱۲:۳۲

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 واکشی میکنید اکانتی که میخوایم باید همنام یا شبیه آدرس سایت(SiteUrl) باشه و طبیعتا من وقتی توی گوگل آنالیز اکانت و پروفایلهام رو میسازم نام دلخواه میدم، شما بجای استفاده از آی دی از اسم استفاده کردید که از روی اسم آی 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 پراپرتی هم به همین صورت.

نویسنده: س محمدرضا برنتی
تاریخ: ۱۳۹۳/۱۱/۲۵ ۱۹:۲۰

با توجه به تحریم بودن این سرویس در ایران شما چه سرویس جایگزینی را پیشنهاد می‌کنید؟

عنوان: **بهینه سازی فایلهای js و css در برنامه‌های ASP.NET با استفاده از Combres - قسمت اول**
نویسنده: سامان هاشمی
تاریخ: ۱۳۹۳/۰۲/۲۸ ۱۸:۴۵
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, MVC, Combres, Minification

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 افزونه‌ی firebug درخواست‌هایی را که از سرور شده‌است، بررسی می‌کنیم. مشاهده خواهید کرد که به ازای هر فایل، یک درخواست به سرور ارسال شده و هیچکدام از فایل‌ها توسط وب سرور فشرده سازی نشده‌اند 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، روشهای گوناگونی وجود دارد که امروز قصد داریم این کار را توسط کتابخانه Combress انجام دهیم ! **نصب کتابخانه Combres**

شما می‌توانید با استفاده از nuget این کتابخانه را به پروژه خود اضافه کنید. **ایجاد فایل تنظیمات**

پس از نصب کتابخانه، فایلی با نام combres.xml در فولدر app_data ساخته می‌شود که تمامی فعالیت‌های کتابخانه براساس آن انجام می‌شود 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" />
```

```
      <resource path="~/scripts/anotherJs.js" />
      <resource path="~/scripts/yetAnotherJs.js" />
    </resourceSet>
  </resourceSets>
</combres>
```

ResourceSet : با استفاده از هر ResourceSet می‌توانید آن مجموعه فایل را در یک درخواست از سرور دریافت کنید.

پارامتر **url** : برای تولید لینک فایل‌ها از آن استفاده میکند.

پارامتر **defaultDuration** : این عدد به تعداد روزهای پیشفرض برای کش کردن فایلها اشاره میکند.

پارامتر **defaultVersion** :در صورتی که مقدار آن auto باشد به ازای هر تغییر، آدرس جدیدی برای فایل موردنظر ایجاد میشود.

پارامتر **defaultDebugEnabled** :با استفاده از آن میتوانید debug mode را مشخص کنید. در صورتیکه مقدار آن auto باشد، این مقدار مستقیما از وب‌کانفیگ خوانده میشود.

مقادیر پیش فرض برای تمامی ResourceSetها استفاده می‌شود و در صورت نیاز میتوان این مقادیر را برای هر ResourceSet بازنویسی کرد. فیلترها برای اعمال تغییراتی در فایل js و CSS استفاده می‌شوند که باید به این شکل معرفی شوند. در قسمت بعد با فیلترها بیشتر آشنا میشویم.

فایل cobmres.xml را به منظور استفاده در پروژه به صورت زیر تغییر می‌دهیم.

```
<?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 می‌شود؛ 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 Route مربوط به Combres را به RouteTable اضافه کنیم. در صورتیکه از نیوگت استفاده کرده باشید، کلاسی به فولدر app_start اضافه شده است که با استفاده از WebActivator اینکار را در Application_Start انجام میدهد؛ در غیر اینصورت باید به صورت زیر این 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 متد CombresLink ارسال میشود. پس از اجرای مجدد برنامه و با استفاده از firebug خواهیم دید که به ازای هر **Set** Resource ، 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.

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:
,textarea{max-width:280px}.field-validation-error{color:#b94a48}.field-validation-valid{
}input.input-validation-error{border:1px solid #b94a48}input[type="checkbox"].input-vali
{border:0 none}.validation-summary-errors{color:#b94a48}.validation-summary-valid{displa
```

GET /combres.axd/siteJs/119039392	200 OK	localhost:55439	67.5 KB
GET site.css	200 OK	localhost:55439	463 B

Headers | Response | Cache

```
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;
}
```

در قسمت بعد با سازوکار combres و روش استفاده از فیلترها، بیشتر آشنا میشویم.

CombresProject-67bd3b7299f24c5484edc35537fa990b.zip

نظرات خوانندگان

نویسنده: آرایه
تاریخ: ۱۳۹۳/۰۲/۲۹ ۹:۴۰

ضمن تشکر به خاطر توضیحات خوب، مزیت استفاده از Combres نسبت به امکان bundle و minification خود ASP.NET MVC چیست؟

نویسنده: سامان هاشمی
تاریخ: ۱۳۹۳/۰۲/۳۰ ۱۳:۴۴

شاید عمده مزیت‌های Combres این چهار مورد زیر باشد:
1- از کتابخانه‌های مختلفی برای بهینه سازی پشتیبانی میکند مانند Microsoft Ajax Minifier و YUI Compressor for .NET و Google Closure Compiler.
2- تعریف جداگانه اطلاعات کش به ازای هر فایل
3- فیلترها
4- لاگ کردن پردازشهای داخلی(به صورت پیشفرض توسط Log4Net انجام میگیرد)

نویسنده: rezaei
تاریخ: ۱۳۹۳/۰۲/۳۰ ۱۷:۳۶

با سلام؛ من تمامی مراحلی که فرمودید را انجام دادم. اما موقع اجرا خطای زیر رو می‌گیره

```
System.ArgumentException: Illegal characters in path
```

```
@Url.CombresLink("jquery")
```

نویسنده: سامان هاشمی
تاریخ: ۱۳۹۳/۰۲/۳۱ ۷:۵۷

احتمالا تو 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: ۱۳۹۳/۰۳/۱۳ ۱۱:۵۶

برای MVC از Combres.Mvc استفاده کنید و سپس از دستور

```
@Html.Raw(WebExtensions.CombresLink("siteCss"))
```

نویسنده: rezaei
تاریخ: ۱۳۹۳/۰۶/۱۱ ۲۳:۱۳

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

ممنون میشم راهنمایی کنید

عنوان: **استفاده از pjax بجای ajax در ASP.NET MVC**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۳/۰۳ ۱۲:۴۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, jQuery, jQuery Ajax, SEO

عموما از ajax برای ارائه سایت‌هایی سریع، با حداقل ریفرش و حداقل مصرف پهنای باند سرور، استفاده می‌شود. اما این روش، مشکلات خاص خود را نیز دارا است. عموما محتوای پویای بارگذاری شده، سبب تغییر آدرس صفحه‌ی جاری در مرورگر نمی‌شود. برای مثال اگر قرار است چندین برگه در صفحه به صورت ajax ایی بارگذاری شوند، تغییر سریع محتوا را مشاهده می‌کنید، اما خبری از تغییر آدرس جاری صفحه در مرورگر نیست. همچنین روش‌های ajax ایی عموما SEO friendly نیستند. زیرا اکثر موتورهای جستجو فاقد پردازشگرهای جاوا اسکریپت می‌باشند و محتوای پویای ajax ایی را مشاهده نمی‌کنند. برای آدرس دهی این مشکلات مهم، افزونه‌ای به نام pjax طراحی شده‌است که کار آن دریافت محتوای HTML ایی از سرور و قرار دادن آن در یک جایگاه خاص مانند یک div است. در پشت صحنه‌ی آن از jQuery ajax استفاده شده، به همراه push state

pjax = pushState + AJAX

Push state API همان HTML5 History API است؛ به این معنا که هرچند 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، layout یا master page یک سایت را درنظر بگیرید. به ازای مرور هر صفحه، یکبار باید تمام قسمت‌های تکراری layout از سرور بارگذاری شوند. توسط pjax به سرور اعلام می‌کنیم، ما تنها نیاز به body صفحات را داریم و نه کل صفحه را.
همچنین اگر مرورگر از جاوا اسکریپت استفاده نمی‌کند، لطفا کل صفحه را همانند گذشته بازگشت بده. به علاوه مسایل سمت کلاینت مانند تغییر آدرس مرورگر و تغییر عنوان صفحه نیز به صورت خودکار مدیریت شوند. این تکنیک را دقیقا 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 ایی و دیگری برای حالت بارگذاری کامل صفحه. حالت ajax ایی آن تنها از رندرکردن 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 پوشه‌ی 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 مساوی pjaxContainer قرار گرفته و از آن در فراخوانی افزونه‌ی pjax استفاده شده‌است. 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 گرفته‌است. 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 نکرده‌است) یا از layout سبک‌تر 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 متد و دو View متناظر با آن ایجاد کنید.
سپس View متد Index را به نحو ذیل تغییر دهید:

```
@{
ViewBag.Title = "Index";
}
```

```
<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 شده‌است. فقط در ویژگی‌های html آن، یک ویژگی جدید به نام withpjax را نیز اضافه کرده‌ایم 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:

همانطور که ملاحظه می‌کنید، با کلیک بر روی لینک About، یک درخواست pjax ایی به سرور ارسال شده‌است؛ به همراه هدرهای ویژه آن. هنوز قسمت‌های اصلی layout سایت مشخص هستند (و مجددا از سرور درخواست نشده‌اند). آدرس صفحه عوض شده‌است. به علاوه قسمت body آن تنها تغییر کرده‌است.

این مثال را از اینجا نیز می‌توانید دریافت کنید

PajxMvcApp.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: رضایی
تاریخ: ۱۳۹۳/۰۳/۰۳ ۱۶:۲۳

با سلام؛ آیا pjax توسط کلیه مرورگرها پشتیبانی میشه؟ مخصوصا IE ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۳/۰۳ ۱۷:۱

- فقط در مرورگرهایی پشتیبانی می‌شود که push state را پیاده سازی کرده باشند: لیست کامل آن‌ها
- اگر مرورگری history.pushState API را پشتیبانی نکند، بارگذاری صفحات آن معمولی خواهند بود (شبیه به حالت بارگذاری کامل برای موتورهای جستجو؛ بدون از کار افتادن برنامه).

نویسنده: MD
تاریخ: ۱۳۹۳/۰۳/۰۴ ۲۳:۵۶

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 View State خاص خودش را خواهد داشت (بر اساس کنترل‌هایی که دارد)، تزریق محتوای آن داخل یک صفحه‌ی دیگر سبب تخریب View state جاری و از پیش موجود می‌شود. در نتیجه امکان ارسال اطلاعات به سرور را با پیام view state is corrupted از دست خواهید داد.

نویسنده: رضایی
تاریخ: ۱۳۹۳/۰۳/۰۵ ۱۳:۵

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 جاوااسکریپتی، به این مطلب مراجعه کنید: « نحوه استفاده از افزونه Firebug برای دیباگ برنامه‌های ASP.NET مبتنی بر jQuery »

نویسنده: رضایی
تاریخ: ۱۳۹۳/۰۳/۰۵ 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 متد post یک اکشن رو فراخوانی می‌کند لینک کار نمی‌کند. مثلا 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: ۱۳۹۳/۰۳/۰۵ ۱۵:۱

- المان‌هایی که به صورت پویا به صفحه اضافه می‌شوند، تحت کنترل مجدد jQuery نخواهند بود، مگر اینکه از متدهای live (منسوخ شده) و یا on استفاده شود. مطابق مستندات این کتابخانه (انتهای صفحه)، متد on به صورت پیش فرض، برای کارکرد مجدد pjax، اعمال می‌شود. بنابراین فقط باید بررسی کنید که آیا نگارش jQuery مورد استفاده، از متد 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 متد on استفاده کنند. مثلا 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: ۱۳۹۳/۰۳/۱۶ ۲۱:۱۲

همه چیز خوبه؛ اما وقتی توی گرید میاییم میزنیم Edit و میریم صفحه Edit و گزینه Save رو میزنیم، برمیگردیم به صفحه Index در حالی که لینک تغییر نکرده و لینک مثلاً اینطوری مونده

www.test.com/edit/1

و وقتی میزنیم گرید به صفحه 2 بره ارور میده که اصلاً همچین صفحه ای 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 ننویسه؟

localhost:18162/Tbl1
Application name
Tbl1
Home
About
Contact
صفحه Index
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
Page 1 of 1
10 items per page
1 - 1 of 1 items

localhost:18162/Tbl1/Edit/2080
Application name
Tbl1
Home
About
Contact
صفحه
Edit
Edit
tbl1
Name
a
Family
444
Age
12
Salary
Coment
Save
Back to List

localhost:18162/Tbl1/Edit/2080
Application name
Tbl1
Home
About
Contact
بازگشت به صفحه
Index
پس از
Edit
و عدم تغییر لینک
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
Edit
Details
Delete
a
444
12
Page 1 of 1
10 items per page
1 - 1 of 1 items

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۳/۱۶ ۲۱:۲۵

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 گرید 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: ۱۳۹۳/۰۳/۱۶ ۲۱:۳۸

نمی دونم Pjax این مشکلا رو چطوری بوجود میاره، اما از وقتی ازش استفاده کردم، توی Save کردن در صورتی که از گزینه‌های خود یک Textbox که خود Browser پیشنهاد میده استفاده بشه، چندتا چندتا ذخیره میشه:

و اینم خروجی یک Save

نویسنده: رمزینه
تاریخ: ۱۳۹۳/۰۳/۱۶ ۲۱:۴۳

ممنون از جوابتون اما مشکل من همینجاست که اصلاً به هیچ خصوصیتی ندادم

```
<input type="submit" value="Save" />
```

اینو گذاشتم اما چون توی

```
@Ajax.beginform(...)
```

هست و توی

```
<div id="pjaxContainer">
  @RenderBody()
</div>
```

که توی layout هست داره بصورت pjax عمل میکنه.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۳/۱۶ ۲۲:۲

- 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 ایی است یا Pjax ایی). اطلاعات بیشتر
- کار pjax فقط ارائه محتوای صفحات است. اگر فعال هم نباشد، برنامه بدون مشکل کار می‌کند و صفحات آن نمایش داده خواهند شد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۳/۱۶ ۲۲:۱۳

با عکس نمی‌شود یک برنامه را از راه دور دیباگ کرد .

نویسنده: احسان شیروان
تاریخ: ۱۳۹۳/۰۳/۱۸ ۱۱:۳۹

تصور بنده بر اینه که شما باید برای داشتن فرم همون Html.BeginForm معمولی رو 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: ۱۳۹۳/۰۳/۱۹ ۰:۵۳

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 هیچکدوم استفاده نکنم نمیدونم چطوری اطلاعات صفحه رو به یه Action توی کنترلر با دکمه Submit ارسال کنم!

نویسنده: فخاری
تاریخ: ۱۳۹۳/۰۷/۱۹ ۱۷:۱۸

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: ۱۳۹۳/۰۷/۱۹ ۱۷:۴۱

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: ۱۳۹۳/۰۷/۲۰ ۱۲:۲۰

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 مفصلا مطرح شده
دانلود مثال
خواستم ببینم که مشکل از چی می‌تونه باشه؟

نویسنده: فخاری
تاریخ: ۱۳۹۳/۰۸/۰۵ ۱۰:۴۰

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: ۱۳۹۳/۰۸/۰۵ ۱۰:۴۵

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: ۱۳۹۳/۰۸/۰۵ ۱۱:۶

این خطا برای یه لحظه به من نمایش داده می‌شود و بعد کل خطاها پاک می‌شود و از نو صفحه بارگذاری می‌شود.
راهی هست که در زمان نمایش خطا به 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: ۱۳۹۳/۰۸/۰۵ ۱۱:۲۵

internal server error صرفا به معنای بروز استثنایی در کدهای سمت سرور شما است. برای لاگ کردن دقیق ریز جزئیات آن‌ها از ELAMH استفاده کنید. نسخه‌ی ساده شده‌ی آن برای ASP.NET MVC در اینجا

نویسنده: فخاری
تاریخ: ۱۳۹۳/۰۸/۰۶ ۱۲:۱

با سلام
از کمک شما ممنون
بالاخره خطا 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 مثلا main در کد زیر را جایگذاری کنه؟

```
<div id="main">
            @RenderBody()
        </div>
```

```
//**********
@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("ارتباط با ما","Contact", "Home"
                                        , null,new { withpjax="with-pjax" })
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۸/۰۶ ۱۲:۲۹

یعنی فایل _PjaxLayout.cshtml هم نیاز به یک سری تعاریف section را دارد؛ مانند:

```
@RenderSection("Scripts", false)
```

کلا View ایی که قرار است رندر شود، اگر دارای تعاریف section اختصاصی هست، باید معادل آن‌ها در فایل layout متناظر، تعریف RenderSection وجود داشته باشد.

نویسنده: شقایق اشتری
تاریخ: ۱۳۹۳/۰۸/۱۹ ۱۲:۱۳

با سلام.
من وقتی از Pjax در یک View که هیچ Layout یی ندارد استفاده می‌کنم به خوبی جواب می‌دهد و صفحه رفرش نمی‌شود .(قسمت پیج بندی 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: ۱۳۹۳/۰۸/۱۹ ۱۲:۲۳

مراجعه کنید به توضیحات قسمت «ب) تغییر فایل ViewStart برنامه ». اگر Layout یک View ذکر نشود، اطلاعات آن‌را از ViewStart دریافت می‌کند. اگر آن‌را صریحا ذکر کنید، همان کاری که در ViewStart برای تشخیص هدر Request.Headers["X-"] PJAX انجام شده، در این حالت باید به صورت دستی انجام و اضافه شود.

نویسنده: شقایق اشتری
تاریخ: ۱۳۹۳/۰۸/۱۹ ۱۲:۳۵

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 پیج بندی لیست اطلاعاتم فقط در این view میخوام استفاده کنم.
اما من میخوام وقتی view خود را در Layout میگذارم جواب دهد.

لیست محصولات من در یک Partial قرار دارد که این Partial هم در View هست .

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۸/۱۹ ۱۳:۵۴

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 خطایی مشاهده نمی‌شود ؟
- آیا header مربوط به X-PJAX در درخواست ارسالی به سرور مطابق تصاویر فوق وجود دارند؟
- آیا در سمت سرور بر اساس هدر X-PJAX دریافتی، فایل layout صحیحی تنظیم می‌شود؟

عنوان: استفاده از افزونه‌ی jsTree در ASP.NET MVC
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۳/۱۹ ۷:۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, jQuery

jsTree یکی از افزونه‌های بسیار محبوب jQuery 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.

دریافت افزونه‌ی jsTree

برای دریافت افزونه‌ی jsTree می‌توان به مخزن کد آن در Github مراجعه کرد و همچنین مستندات آن‌را در سایت jstree.com قابل مطالعه هستند.

تنظیمات مقدماتی jsTree

در این مطلب فرض شده‌است که فایل jstree.min.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; } // نام این خواص باید با مستندات هماهنگ 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; }
    }
```

```
    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 اسامی 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 دقیقا معادل id یک رکورد در بانک اطلاعاتی باشد. Text عنوان گره‌ای (node) است که نمایش داده می‌شود. icon 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 کاملا 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 فرمت 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 از 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 کرده‌ایم 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 کرده‌ایم.
- 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، عناوینی مانند rename، create و امثال آن نمایش داده می‌شوند که به نحو فوق می‌توان آن‌را 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 مرتبط با delete_node کار حذف یک گره اطلاع رسانی می‌شود. create_node مربوط است به ایجاد یک گره یا زیر شاخه‌ی جدید. rename_node پس از تغییر نام یک گره فراخوانی خواهد شد. move_node مربوط است به کشیدن و رها کردن یک گره در یک زیر شاخه‌ی دیگر. copy_node برای copy/paste یک گره تعریف شده‌است. Changed یک callback عمومی است. dblclick برای عکس العمل نشان دادن به رخداد دوبار کلیک کردن بر روی یک گره می‌تواند بکار گرفته شود. select_node با انتخاب یک گره 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 گره، متن تغییر یافته آن، والد گره و بسیاری از مشخصات دیگر را دریافت و ثبت کرد. پس از تعریف متد postJsTreeOperation فوق، آن‌را باید به callbackهایی که پیشتر معرفی شدند، 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 متد postJsTreeOperation، یک Operation خاص را مانند CreateNode (تعریف شده در enum ایی به نام JsTreeOperation در سمت سرور) به همراه data، به سرور post می‌کند.
و معادل سمت سرور دریافت کننده‌ی این اطلاعات، اکشن متد ذیل می‌تواند باشد:

```
[HttpPost]
        public ActionResult DoJsTreeOperation(JsTreeOperationData data)
        {
            switch (data.Operation)
            {
                case JsTreeOperation.CopyNode:
                case JsTreeOperation.CreateNode:
                    //todo: save data
                    var rnd = new Random(); // 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 شده‌است:

```
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 دقیقا با اعضای شیء جاوا اسکریپتی که متد postJsTreeOperation به سمت سرور ارسال می‌کند، تطابق دارد.
در اینجا Href را نیز مشاهده می‌کنید. همانطور که عنوان شد، اعضای JsTreeNodeAAttributes 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.

چند نکته‌ی تکمیلی

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 مرتبط با select_node می‌توان به گره انتخابی درستی یافت. سپس می‌توان این گره را در callback متناظر با 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":fals
e},"children":[{"id":"2","text":"آنات","icon":"/Content/images/nuclear.png","state":{"opened":true,"dis
abled":false,"selected":false},"children":[{"id":"4","text":"آموزش","icon":"/Content/images/nuclear.pn
g","state":{"opened":true,"disabled":false,"selected":false},"children":[],"li_attr":{"data":null},"a_a
ttr":{"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}}],"li_attr":{"data":null},"a_attr":{"href":
null}},{"id":"1","text":"پرنیان","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":{"hre
f":null}},{"id":"3","text":"پشتیبانی","icon":"/Content/images/nuclear.png","state":{"opened":true,"disab
led":false,"selected":false},"children":[],"li_attr":{"data":null},"a_attr":{"href":null}},{"id":"2","t
ext":"فروش","icon":"/Content/images/nuclear.png","state":{"opened":true,"disabled":false,"selected":fa
lse},"children":[],"li_attr":{"data":null},"a_attr":{"href":null}}],"li_attr":{"data":null},"a_attr":{"
href":null}}],"li_attr":{"data":null},"a_attr":{"href":null}}]
```


منتهی همش خطای too much recursion میگیرم، در صورتی که حلقه ای رو هم نمی‌بینم که ایجاد شده باشه داخل رشته json تولید شده.

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: ۱۳۹۳/۰۵/۰۷ ۲:۳۳

گره‌های تعریف شده unique ID ندارند. این unique ID در کل tree معنا پیدا می‌کند و الزاما ارتباطی به ID رکورد شما در یک جدول خاص بانک اطلاعاتی ندارد.

نویسنده: ناصر پورعلی
تاریخ: ۱۳۹۳/۰۵/۱۱ ۱۶:۱۳

ممنون، مشکل قبلی حل شد.
مشکل دیگه ام اینه که من مجبورم از jquery 1.6.1 استفاده کنم، آیا نسخه ای از jstree هست که با این نسخه jquery به خوبی کار کنه؟ ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۱۱ ۱۸:۲۸

از jQuery>=1.9 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 کرده. بنابراین افزونه‌های به 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 jQuery migrate است:

```
<script src="jquery.js"></script>
<script src="jquery-migrate-1.2.1.js"></script>
```

نویسنده: سعیده
تاریخ: ۱۳۹۳/۰۷/۲۴ ۱۵:۴۶

با سلام و تشکر از آموزشتون.

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 متد 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.

عنوان: **تفاوت ViewData و ViewBag و TempData و Session در MVC**
نویسنده: م منفرد
تاریخ: ۱۳۹۳/۰۳/۲۳ ۱۹:۱۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

در MVC راه‌های متفاوتی برای ارسال اطلاعات از controller به view و در redirectها وجود دارد. در این مقاله سعی شده تفاوت‌های آنها به صورت مختصر نمایش داده شود. این مقاله در حقیقت یک ترجمه آزاد از این مقاله است که امیدوارم خوب ترجمه شده باشد. **ViewData** :
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 c# 4 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 TempDataDictionary 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 requestهای 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 میشه. چیکار کنم مقدارش رو حفظ کنه.

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۳/۲۵ ۱۸:۱۱

می‌تونید 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

مطلب خوب و جالبی بود، کاش تصویر موجود در مقاله منبع رو توی مقاله تون میاوردین، به نظرم خیلی تصویر گویا و جالبی هست

عنوان: **استفاده از چند فرم در کنار هم در ASP.NET MVC**
نویسنده: مرتضی دلیل
تاریخ: ۱۳۹۳/۰۳/۲۸ ۱۷:۳۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

اجرای این نوع صفحات کار سختی نیست؛ با کمی جستجو در اینترنت مثلا در اینجا میتوانید چیزهای خوبی پیدا کنید. اما متاسفانه اکثر مثال‌ها چیزی شبیه قرار دادن پارشال "ورود اعضا" در کنار پارشال "ثبت نام" هستند. حتما متوجه شده‌اید که معمولا 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:

اولا : 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. مثلا 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:

ViewModelهای مرتبط با این دو بخش به شکل زیر هستند :

ContactVM .cs

```
public class ContactVM
    {
        [Display(Name = "نام")]
        [Required(ErrorMessage = "لطفا {0} را وارد کنید")]
        public string Name { get; set; }

              [EmailAddress(ErrorMessage = "آدرس ایمیل صحیح نیست")]
        [DataType(DataType.EmailAddress)]
        [Display(Name = "آدرس ایمیل")]
        [Required(ErrorMessage = "لطفا {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 = "لطفا {0} را وارد کنید")]
        [Display(Name = "حاصل جمع")]
        public string Captcha { get; set; }
    }
```

SubscriberVM .cs

```
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")] /*.Net4.5*/
        [Display(Name = "ایمیل")]
        [Required(ErrorMessage = "لطفا {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()
        {
```

```
            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" />
        <img alt="Captcha" id="imgcpatcha" src="@Url.Action("CaptchaImage","Captcha")" />
    </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 tml_

```
@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-
```

```
right", @id = "email", @name = "email", placeholder = "ایمیل", @style = "direction: ltr;width: 50%",
@required = "required" })
       @*     <button type="submit" name="submitValue">ثبت ایمیل</button>*@
            <input type="submit" value="ثبت ایمیل" name="submitValue" />
        </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 متدی را برای صدا زدن در این بخش در نظر بگیرید، هنگام Postback به مشکل برخورد خواهید کرد؛ چون آدرس آن اکشن متد به شکل صریح در آدرس مرورگر فراخوانی میشود و پارشال ما پس از Post-back به تنهایی و بدون Layout نمایش داده خواهد شد. اسم بردن از اکشن متد وقتی کارساز است که آن اکشن متد قرار باشد یک Redirect انجام دهد ولی هدف ما این است که صفحه را از دست ندهیم و پیام‌های خطای ModelState را در همان صفحه قبل و پس از Post-back ببینیم و همچنین پس از انجام عملیات (مثلا ارسال پیام) همین صفحه نمایش داده شود.

نکته دوم : نکته اول یک مشکل دارد! اگر به شکل صریح اکشن متد مربوط به Post-back مشخص نشود، بطور اتوماتیک تمامی اکشن متدهایی که ویژگی [HttpPost] دارند اجرا خواهند شد. این یعنی هر دو اکشن متد Contact و Subscriber اجرا می‌شوند و بنابر آنچه در اکشن متدها نوشته‌ایم هر دو ModelState بررسی می‌شود که این هدف ما نیست. مثلا فرم سمت چپ را تکمیل کرده ایم و دکمه "ثبت ایمیل" را فشار داده‌ایم و صفحه Postback می‌شود و با اینکه ایمیل در بانک ثبت شده اما فرم سمت راستی با خطا ظاهر میشود که چرا فیلدها خالی هستند!؟

برای حل این مشکل کافیست خاصیت name مربوط به دکمه‌ها را به شکل یک ورودی برای اکشن متدها بفرستیم و بر اساس وضعیت آن تنها state 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 پراپرتی‌های با نام یکسان نداشته باشند. مثلا پراپرتی 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 متد مربوط به آن (مثلا ارسال ایمیل، ثبت در بانک یا ...) در صورتی که عملیات با موفقیت انجام شد حتما ModelState را clear کنید. با اینکار پس از Post-back فیلدهای پارشال‌ها خالی میشوند.

نکته پنجم : میتوانید به سادگی مدیریت خطا را به کمک جی کوئری انجام دهید؛ مثلا فرض کنید میخواهیم اگر ایمیل کاربر برای دریافت خبرنامه با موفقیت ثبت شد، پیامی مبنی بر موفقیت برای وی بفرستیم؛ اکشن متد HttPost 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-dismissable">
            <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 استرپِ 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 متد برای Html.BeginForm تعریف کنیم تا این دردسرها را نداشته باشیم.

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.

نظرات خوانندگان

نویسنده: میثم ثوامری
تاریخ: ۱۳۹۳/۰۳/۲۸ ۲۳:۵

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: ۱۳۹۳/۰۳/۲۹ ۰:۳۳

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 جاهایی ساختن پارشال یا اکشن هایی 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 تصورش مسخره باشه ولی شاید یه روزی به کار اومد. حداقلش یه بازی خوب هست برای فهمیدن ظرفیت‌های MVC

نویسنده: رمزینه
تاریخ: ۱۳۹۳/۰۳/۲۹ ۱۱:۱۳

مرسی
من دقیقاً الان دنبال همین موضوع بودم و یه ویو دارم با 9 پارشیال ویو که از همین روش باید استفاده کنن.

نویسنده: علیرضا همتی
تاریخ: ۱۳۹۳/۰۳/۳۰ ۱۲:۱۱

سلام.
مرسی از مقاله خوبتون.
من یه مشکلی دارم شاید اگه ممکنه منو هم راهنمایی کنین.
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 میشن. 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 می خواستم ببینم پیشنهاد شما چیه. ممنون از همتون.

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۳/۳۰ ۱۲:۱۸

در *حاشیه!* تلفظش پارشال یا پارشل هست: https://www.google.com/search?q=partial

نویسنده: رشوند
تاریخ: ۱۳۹۳/۰۵/۳۰ ۲۰:۵۱

با سلام سپاس؛

1- اگر لطف کنید کمی در باره Html.Action توضیح بدید ممنون میشم.
2- دو تا partial را با استفاده از Html.Action و return اکشن مربوط, به صفحه اصلی اضافه کردید.

```
public ActionResult Contact()
        {
            return PartialView("_Contact", model);
        }
```

model هر کدام از partialها رو مشخص کردید

```
@model IrsaShop.Models.ViewModel.ContactVM
```

```
@model IrsaShop.Models.SubscriberVM
```

سوال: در صفحه اصلی مودل ما به چه صورتی خواهد بود؟

```
@model IrsaShop.Models.ViewModel.؟
```

آیا باید از viewmodel استفاده کنیم که هر دو (SubscriberVM , ContactVM) و موارد مورد نیاز دیگر صفحه را در بر بگیرد؟
با فرض اینکه مثلا خود صفحه هم از مودل دیگری برای استفاده از helper استفاده می‌کند,
آیا مودل برنامه تاثیری بر partialها دارد؟(در این روش)
@Html.Action مودلی را به partial ارسال نمی‌کنه برخلافه @Html.Partial که model مورد نیاز را ارسال می‌کند.

(لطفا 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:
حین 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
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 ۰:۴۰

child action 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 بنده می‌خواهم بعد از پایان صحیح عملیات اکشن دیگه ای فعال بشه , مثلا بعد از ثبت نظر مربوط به یک خبر در سایت , خبر مورد نظر 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(پی اپی 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 child action نبوده. از Html.Action زمانی استفاده میشه که قراره چند partial view از پیش مقدار دهی شده در صفحه، از منابع داده مختلفی مقدار دهی بشن. مثلا قراره در کنار صفحه، از یک منبع داده، آمار سایت رو نمایش بدید (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 view معمولی هست، در ASP.NET MVC مجاز هستید n تا فرم در صفحه قرار بدید. 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

نویسنده: مهدیس
تاریخ: ۱۳۹۳/۱۱/۲۰ ۱۱:۱۵

سلام ممنون از این مقاله خوبتون.
فقط من یه مشکلی دارم. سه تا فرم دارم که توی پارشال ساخته شدن و توسط ویومدل هم مقدار دهی میشن.
ولی در زمان اجرا عمل ذخیره رو که انجام میده بعدش خطای زیر رو میده:
Error executing child request for handler

''System.Web.Mvc.HttpHandlerUtil+ServerExecuteHttpHandlerAsyncWrapper.

لطف میکنید راهنمایی کنید.

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۱۱/۲۰ ۱۲:۵۱

بدون دیدن کدهای شما نمیشه نظر داد . آیا در کنترلری که دارید از HttpGet روی اکشن متدها استفاده کردید؟ حذفش کنید بعد دوباره تست کنید. آیا روی اکشن متد از NonAction بجای ChildActionOnly استفاده کردید؟ کلا ویژگی‌های اضافه شده را باید دقیقا بررسی کنید.

نویسنده: مهدیس
تاریخ: ۱۳۹۳/۱۱/۲۰ ۱۳:۳

ممنون از اینکه پاسخ دادید.

خیر از Httpget و NonAction استفاده نکردم ولی از ChildActionOnly بالای اکشنها استفاده کردم
در کنترلر به صورت زیر اکشنها رو نوشتم:

```
[Authorize(Roles = "Administrator")]
        public ActionResult AddUserRole()
        {
            return View();
        }

         ////////////////////////////////////////////////////////

        //اضافه کردن نقش////////////
        [Authorize(Roles = "Administrator")]
        [HttpGet]
        [ChildActionOnly]

        public ActionResult AddRole()
        {

            return PartialView("PartialRole");
        }

        [ChildActionOnly]
        [HttpPost]
        //[ValidateAntiForgeryToken]
        public ActionResult AddRole(tblRole rl, string submitValue)
        {
            if (submitValue == "ذخیره نقش")
            {
                //تبدیل تاریخ میلادی به شمسی
                rl.RoleDateCreate = Utility.ToPersianDate(DateTime.Now);

                UserRepository user = new UserRepository();
                //برای وارد کردن خودکار شناسه کاربر وارد شده
                rl.UserIDCreate_FK = Convert.ToInt64(user.FindUserID(User.Identity.Name));

                if (ModelState.IsValid)
                {
                    RoleRepository acr = new RoleRepository();
                    var model = acr.Add(rl);
                    ViewBag.Message = "نقش با موفقیت ثبت شد";
                    ModelState.Clear();
                }
            }
            else
            {
                ModelState.Clear();
            }
            return PartialView("PartialRole", rl);

        }
        ////////////////////////////////////////////////////////
```

```
        //اضافه کردن کاربر//////////
        [Authorize(Roles = "Administrator")]
        [HttpGet]
        [ChildActionOnly]

        public ActionResult AddUser()
        {
            return PartialView("PartialUser");
        }

        [ChildActionOnly]
        [HttpPost]

        public ActionResult AddUser(tblUser rl, string submitValue)
        {
            if (submitValue == "ثبت نام")
            {
                if (ur.ExistUserName(rl.UserName))//برای اینکه نام کاربری تکراری نباشد
                {
                    ViewBag.Warning = "این نام کاربری تکراری میباشد";

                }
                else
                {

                    //برای وارد کردن خودکار شناسه کاربر وارد شده
                    rl.UserIDCreate = Convert.ToInt64(ur.FindUserID(User.Identity.Name));

                    //تبدیل میلادی به شمسی
                    rl.UserDateCreate = Utility.ToPersianDate(DateTime.Now);

                    if (ModelState.IsValid)
                    {
                        UserRepository acr = new UserRepository();
                        var model = acr.Add(rl);
                        ViewBag.Message = "کاربر با موفقیت ثبت شد";
                        ModelState.Clear();
                    }
                }
            }
            else
            {
                ModelState.Clear();
            }
            return PartialView("PartialUser", rl);

        }

        ////////////////////////////////////////////////////////////

        // //////دادن نقش به کاربر

        [ChildActionOnly]

        [HttpGet]
        [Authorize(Roles = "Administrator")]
        public ActionResult RoleOfUser()
        {
            var blUser = new UserRepository();
            var blRole = new RoleRepository();
            var model = new RoleUserViewModel();

            model.username = blUser.Select().ToList();
            model.rolename = blRole.Select().ToList();

            return PartialView("PartialRoleOfUser", model);
        }

        [ChildActionOnly]
        [HttpPost]
        public ActionResult RoleOfUser(tblRoleUser rl, string submitValue)
        {
            if (submitValue == "ذخیره نقش کاربر")
            {
                //برای وارد کردن خودکار شناسه کاربر وارد شده
                rl.UserIDCreate_FK = Convert.ToInt64(ur.FindUserID(User.Identity.Name));

                //تبدیل میلادی به شمسی
                rl.RoleUserDateCreate = Utility.ToPersianDate(DateTime.Now);

                if (ModelState.IsValid)
```

```
                {
                    RoleUserRepository acr = new RoleUserRepository();
                    var model = acr.Add(rl);
                    ViewBag.Message = "نقش کاربر با موفقیت ثبت شد";

                    ModelState.Clear();
                }
            }
            else
            {
                ModelState.Clear();
            }
            return PartialView("PartialRoleOfUser", rl);

        }
```

در پارشیال ویو Role

```
@model MeterControl.ViewModels.RoleUserViewModel
<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>

@using (Ajax.BeginForm("AddUserRole", "Home", new AjaxOptions { HttpMethod = "Post", Url =
"/Home/AddUserRole" })){
    <fieldset>
        <legend>tblRole</legend>



        <div>
            @Html.LabelFor(model => model.role.RoleName)
        </div>
        <div>
            @Html.EditorFor(model => model.role.RoleName)
            @Html.ValidationMessageFor(model => model.role.RoleName)
        </div>



        @ViewBag.Message
        <p>
            <input type="submit" value="ذخیره نقش" name="submitValue" />
        </p>
    </fieldset>
}
```

و در ویو کلی هم به صورت زیر اکشنها صدا زده شده اند:

```
@{
    ViewBag.Title = "AddUserRole";
}

<h2>AddUserRole</h2>
<div>
    <div>
        @Html.Action("AddRole", "Home")
    </div>
    <div>
        @Html.Action("AddUser", "Home")
        @Html.Partial("PartialUser")
    </div>
    <div>
        @Html.Action("RoleOfUser","Home")
    </div>
</div>
```

به خط زیر خطا میدهد

```
@Html.Action("AddRole", "Home")
```

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۱۱/۲۰ ۱۳:۱۵

[HttpGet] را از روی AddRole اول حذف کنید.

ضمنا زمانیکه فقط قصد نمایش یک Partial view را دارید، نیازی نیست حتما از Html.Action استفاده کنید؛ چون سربار بالایی دارد. هر بار فراخوانی Html.Action سبب راه اندازی مجدد سیکل پردازشی کنترلر جاری میشه. اگر فقط قصد نمایش ساده این‌ها است (بدون پاس دادن اطلاعات اضافه‌تری به آن‌ها)، از همان Html.Partial استفاده کنید. برای مثال در کدهای فوق سه مورد AddUser ، AddRole و RoleOfUser اضافی هستند. حذفشون کنید و این‌ها را با RenderPartial داخل View تعویض کنید. اون مورد RoleOfUser که پارامتر دریافت می‌کنه رو با یک ViewModel اطلاعاتش رو به View برای رندر ارسال کنید.

نویسنده: مهدیس
تاریخ: ۱۳۹۳/۱۱/۲۰ ۱۳:۱۹

حذف کردم ولی هنوز همون خطا رو میده

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۱۱/۲۰ ۱۳:۲۴

برای دیباگ بهتر، زمانیکه صفحه‌ی exception در ویژوال استودیو ظاهر شد، روی دکمه‌ی continue کلیک کنید. بعد صفحه‌ی زرد رنگ ASP.NET ظاهر میشه. اینجا خطای کاملتری نمایش داده میشه با جزئیات بیشتری.

نویسنده: مهدیس
تاریخ: ۱۳۹۳/۱۱/۲۰ ۱۳:۲۶

```
public class RoleUserViewModel
    {
        public tblRole role { get; set; }
        public tblUser user { get; set; }

        public tblRoleUser roleuser { get; set; }

        //برای اینکه در صفحه برای نام کاربری و نام نقش دراپ دان لیست بگذاریم
        public IEnumerable<tblUser> username { get; set; }
        public IEnumerable<tblRole> rolename { get; set; }

    }
```

این هم کد ویومدل هست. که هر سه پارشیال رو با این ویومدل مقدارهی شدن

نویسنده: مهدیس
تاریخ: ۱۳۹۳/۱۱/۲۰ ۱۳:۲۸

بعد از زدن دکمه continue خطایی توی صفحه نمیده و همون صفحه سایت رو نشون میده بدون خطایی

نویسنده: مهدیس
تاریخ: ۱۳۹۳/۱۱/۲۰ ۱۳:۳۳

به جای Html.Action هم نوشتم Html.RenderPartial ولی خطای زیر رو میده:

Server Error in '/' Application.

Compilation Error

Description: An error occurred during the compilation of a resource required to service this request. Please review the following specific error details and modify your source code appropriately.

Compiler Error Message: CS1502: The best overloaded method match for 'System.Web.WebPages.WebPageExecutingBase.Write(System.Web.WebPages.HelperResult)' has some invalid arguments

Source Error:

```
Line 7:  <div class="row footerclass">
Line 8:      <div class="col-md--6">
Line 9:          @Html.RenderPartial("PartialRole")
Line 10:     </div>
Line 11:     <div class="col-md-6">
```

Source File: c:\Users\milad\Desktop\LogDezfouli93.11.04\FinalMeterControl\MeterControl_E2_931120\MeterControl\Views\Home\AddUserRole.cshtml **Line:** 9

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۱۱/۲۰ ۱۳:۳۶

مشکل syntax داره. یکی از دو حالت زیر را استفاده کنید:

```
@{ Html.RenderPartial("......."); }
```

or

```
@{ Html.RenderPartial("......", Model); }
```

نویسنده: مهدیس
تاریخ: ۱۳۹۳/۱۱/۲۰ ۱۳:۴۳

با هر دو مورد خطا میده:

Server Error in '/' Application.

Compilation Error

Description: An error occurred during the compilation of a resource required to service this request. Please review the following specific error details and modify your source code appropriately.

Compiler Error Message: CS1973: 'System.Web.Mvc.HtmlHelper<dynamic>' has no applicable method named 'RenderPartial' but appears to have an extension method by that name. Extension methods c arguments or calling the extension method without the extension method syntax.

Source Error:

```
Line 8:      <div class="col-md--6">
Line 9:
Line 10:         @{ Html.RenderPartial("PartialRole", Model); }
Line 11:     </div>
Line 12:     <div class="col-md-6">
```

Source File: c:\Users\milad\Desktop\LogDezfouli93.11.04\FinalMeterControl\MeterControl_E2_931120\MeterControl\Views\Home\AddUserRole.cshtml **Line:** 10

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۱۱/۲۰ ۱۳:۴۷

اینبار می‌گه مدل رو در این متد جدید استفاده کردی اما بالای صفحه مدلی تعیین نشده و به صورت پیش فرض dynamic هست. بالای صفحه‌ی View، با @model MyViewModel باید مدلی را که قصد دارید پاس بدید، دقیقا مشخص کنید.

نویسنده: مهدیس
تاریخ: ۱۳۹۳/۱۱/۲۰ ۱۴:۰

ممنون از راهنمایی بسیار خوبتون. خیلی کمکم کرد

فقط اینکه الان به کمبوباکس ایراد گرفته و میگه مقداری نداره در صورتیکه پر شده.

نویسنده: مهدیس
تاریخ: ۱۳۹۳/۱۱/۲۰ ۱۴:۱

Server Error in '/' Application.

Object reference not set to an instance of an object.

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.NullReferenceException: Object reference not set to an instance of an object.

Source Error:

```
Line 14:        </div>
Line 15:        <div class="editor-field">
Line 16:            @Html.DropDownListFor(m => m.roleuser.UserID_FK, new SelectList(Model.username, "UserID", "UserName"), ".......")
Line 17:            @Html.ValidationMessageFor(m => m.roleuser.UserID_FK)
Line 18:        </div>
```

Source File: c:\Users\milad\Desktop\LogDezfouli93.11.04\FinalMeterControl\MeterControl_E2_931120\MeterControl\Views\Home\PartialRoleOfUser.cshtml **Line:** 16

Stack Trace:

```
[NullReferenceException: Object reference not set to an instance of an object.]
   ASP._Page_Views_Home_PartialRoleOfUser_cshtml.Execute() in c:\Users\milad\Desktop\LogDezfouli93.11.04\FinalMeterControl\MeterControl_E2_931120\MeterContr
```

نویسنده: مهدیس
تاریخ: ۱۳۹۳/۱۱/۲۰ ۱۵:۸

ببخشید یه سوال دیگه هم داشتم اینکه این RenderPartial هیچ اکشنی رو فراخوانی نمیکنه و با این دستور من هیچ داده ای رو نمیتونم توی دیتابیس ذخیره کنم.

ممنون میشم راهنماییم کنید.

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۱۱/۲۰ ۱۵:۲۰

«هیچ اکشنی رو فراخوانی نمیکنه»

فرقش با قبل فقط همین یک مورد هست. مابقیش فرقی نمی‌کنه. RenderAction کارش فقط رندر هست و نه ثبت چیزی در دیتابیس. من فقط عنوان کردم AddUser() ActionResult اضافی هست و ساده‌تر هم میشه مدیریتش کرد. متد (ActionResult

AddUser(tblUser rl, string submitValue بدیهی هست که باید مانند قبل برای مدیریت Postها باشد.

عنوان: **انجام کارهای پس زمینه در ASP.NET 4.5.2**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۰۲ ۱۰:۲۵
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, Schedule, MVC

دات نت 4.5.2 قابلیت توکاری را به نام در صف قرار دادن یک کار پس زمینه، اضافه کرده‌است که در ادامه خلاصه‌ای از آن‌را مرور خواهیم کرد.

روش متداول 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 اصلا 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 خواهند رفت.

معرفی HostingEnvironment.QueueBackgroundWorkItem

متد HostingEnvironment.QueueBackgroundWorkItem به دات نت 4.5.2 اضافه شده‌است 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 نخواهد 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 حداکثر 30 ثانیه فرصت خواهید 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 async 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 فراخوانی 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 متد HostingEnvironment.RegisterObject می‌توان این کلاس را به موتور ASP.NET معرفی کرد. در این حالت زمانیکه AppDomain قرار است خاتمه یابد، متد Stop اینترفیس IRegisteredObject کار اطلاع رسانی را انجام می‌دهد.
توسط QueueBackgroundWorkItem دقیقا از همین روش به همراه فراخوانی ThreadPool.QueueUserWorkItemجهت اجرای متد معرفی شده‌ی به آن استفاده می‌شود.
از مکانیزم IRegisteredObject در DNT Scheduler نیز استفاده شده‌است.

پیشنیازها

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 را نصب کرده باشید، این نگارش هم اکنون بر روی سیستم شما فعال است. اگر خیر، امکان دریافت و نصب آن، به صورت جداگانه 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 باشید 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 الزاما به این معنا نیست که کار درخواستی شما حتما اجرا خواهد شد. زمانیکه که کار خاتمه‌ی AppDomain آغاز می‌شود، فراخوانی‌های QueueBackgroundWorkItem دیگر پردازش نخواهند 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: ۱۳۹۳/۰۹/۰۸ ۱۵:۱۲

با سلام؛ من می‌خوام یک کار در پس زمینه انجام بدم به صورتی که یک تابع را با چند آرگمان صدا بزنم و این تابع در یک تسک جدید اجرا بشود، به طوری که اگر من این تابع را هر چند بار که صدا بزنم همه آنها در یک تسک جدید و مجزا از هم اجرا بشود.
من چطوری میتونم این کار را انجام بدهم؟ با تشکر.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۰۸ ۱۵:۱۸

- مطلب فوق در همین مورد است. `BackgroundWorkItem` **`Queue`** به معنای در صف قرار دادن این کارها برای پردازش مجزا از هم است.
- از آن استفاده کردید؟ به چه مشکلی برخوردید؟

نویسنده: محمد سعید
تاریخ: ۱۳۹۳/۱۱/۰۸ ۱۶:۱۹

باسلام
من میخوام یک کار در پس زمینه انجام بدم که طول زمان انجامش بیشتر از خیلی بیشتر از سی ثانیه است. از اونجا که برای انجامش فقط سی ثانیه زمان دارم(طبق مطالب متن)، آیا هنوز میتونم از این روش استفاده کنم؟ اگه آره، چطوری؟
پیشاپیش از توجهتون تشکر میکنم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۱/۰۸ ۱۷:۴۷

«برای انجامش فقط سی ثانیه زمان دارم»
خیر؛ اینطور نیست. این مورد فقط مرتبط است به زمانیکه قرار است سرور یا برنامه ری‌استارت شوند (خاتمه‌ی `AppDomain`). در غیر اینصورت محدودیت زمانی ندارید.

عنوان: **نمایش اخطارها و پیام‌های بوت استرپ به کمک TempData در ASP.NET MVC**
نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۳/۰۴/۰۵ ۱۲:۲۰
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 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-dismissable">
  <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>
```

```
  نمایش اعلان
</div>
```

در ادامه قصد داریم این پیام را بعد از ثبت اطلاعات، به کاربر نمایش دهیم. یعنی در داخل کد، امکان 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 Dismissable { get; set; }
}
```

در کلاس فوق خصوصیت یک alert را تعریف کرده‌ایم (از خاصیت TempDataKey جهت پاس دادن alertها به view 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,
```

```
                Dismissable = dismissable
            });

            TempData[Alert.TempDataKey] = alerts;
        }
}
```

از متدهایی که به صورت عمومی تعریف شده‌اند جهت ارسال پیام به view استفاده می‌کنیم. متد AddAlert نیز جهت ایجاد لیستی از پیام‌ها(Alert) مورداستفاده قرار می‌گیرد؛ زیرا ممکن است بخواهید هم زمان از چندین متد عمومی فوق استفاده کنید، یعنی چندین پیام را به کاربر نمایش دهید.

نکته: در کد فوق از TempData جهت پاس دادن شیء alerts استفاده کرده‌ایم. TempData به صورت short-lived عمل می‌کند به دو دلیل: 1- بلافاصله بعد از خوانده شدن، حذف خواهد شد. 2- پس از پایان درخواست از بین خواهد رفت. از TempData جهت پاس دادن داده‌ها از درخواست فعلی به درخواست بعدی (redirect از یک صفحه به صفحه دیگر) استفاده می‌شود. یعنی در زمان redirect سعی می‌کند داده‌های بین redirectها را در خود نگه دارد. اگر از ViewBag و ViewData استفاده می‌کردیم داده‌های داخل آنها بلافاصله بعد از redirect شدن null می‌شدند.
به طور مثال اکشن متد 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 خطاها از TempData استفاده می‌شود.

اکنون در هر کنترلری که می‌خواهید پیامی را به صورت alert، پس از ثبت اطلاعات به کاربر نمایش دهید، باید از کنترلر BaseController ارث‌بری 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)
        {
```

```
            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.Dismissable ? "alert-dismissable" : null;
        <div class="alert alert-@alert.AlertStyle @dismissableClass">
            @if (alert.Dismissable)
            {
                <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>
```

اطلاعات مورد نظر با موفقیت حذف گردید!

نظرات خوانندگان

نویسنده: محمد رعیت پیشه
تاریخ: ۱۳۹۳/۰۴/۰۵ ۲۳:۴۷

آیا تاحالا این روش رو در فرم‌های Modal هم امتحان کردید؟ فکر می‌کنم با یک مشکل برخورد خواهید کرد اونم اینکه پیغام در تمام صفحات زیرین هم نمایش داده خواهد شد، همچنین در Partial Viewها باید مکانیزمی به کار برده شود 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: ۱۳۹۳/۰۴/۰۵ ۲۳:۵۴

- در اینجا هدف نمایش پیام‌ها در صفحه ایی دیگر است.
- پیام‌ها زمانی نمایش داده می‌شوند که TempData حاوی اطلاعات ارسالی باشد. فراخوانی partial نیز یکبار درون Layout انجام می‌شود.

نویسنده: مهدی
تاریخ: ۱۳۹۳/۰۴/۰۶ ۱۱:۴۴

آیا این روش رو میشه برای صفحاتی که بصورت Ajax کار میکنن استفاده کرد؟
این روش شما تنها در صورت Refresh صفحه کار میکنه

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۳/۰۴/۰۶ ۱۳:۵۷

بله؛ می‌توانید به عنوان مثال در viewی createتان 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: ۱۳۹۳/۰۶/۱۶ ۱۷:۱۵

برای مواقعی که سشن رو خاموش کردیم بین ریدایرکت‌ها این روش جواب نمیده آیا راه جایگزینی هست؟

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۳/۰۶/۱۶ ۱۷:۲۱

می‌توانید رفتار TempData را در حالت خاموش بودن سشن با پیاده‌سازی اینترفیس ITempDataProvider تغییر دهید.(+ ، +)

نویسنده: محمد صاحب
تاریخ: ۱۳۹۳/۰۶/۱۶ ۱۷:۴۶

مرسی
با فقط خواندنی کردن سشن هم جواب نداد !

```
[SessionState(SessionStateBehavior.ReadOnly)]
```

میمونه نوشتن پروایدر که باید براش وقت بذارم.

عنوان: صفحه بندی و مرتب سازی خودکار اطلاعات به کمک jqGrid در ASP.NET MVC
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۱۱ ۱۲:۴۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, jqGrid

jqGrid یکی از افزونه‌های بسیار محبوب jQuery جهت نمایش جدول مانند اطلاعات، در سمت کلاینت است. توانمندی‌های آن صرفا به نمایش ستون‌ها و ردیف‌ها خلاصه نمی‌شود. قابلیت‌هایی مانند صفحه بندی، مرتب سازی، جستجو، ویرایش توکار، تولید خودکار صفحات افزودن رکوردها، اعتبارسنجی داده‌ها، گروه بندی، نمایش درختی و غیره را نیز به همراه دارد. همچنین به صورت توکار پشتیبانی از راست به چپ را نیز لحاظ کرده‌است.
مجوز استفاده از فایل‌های جاوا اسکریپتی آن MIT است؛ به این معنا که در هر نوع پروژه‌ای قابل استفاده است. مجوز استفاده از کامپوننت‌های سمت سرور آن که برای نمونه جهت ASP.NET MVC یک سری HTML Helper را تدارک دیده‌اند، تجاری می‌باشد. در ادامه قصد داریم صرفا از فایل‌های JS عمومی آن استفاده کنیم.

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 بیشتر توصیه می‌شود، زیرا به صورت خودکار وابستگی‌های jQuery و همچنین jQuery UI آن‌را نیز به همراه داشته و نصب خواهد کرد.
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 نیز از jQuery UI دریافت می‌کند. بنابراین اگر قصد تغییر قالب آن‌را داشتید تنها کافی است یک قالب استاندارد دیگر jQuery UI را مورد استفاده قرار دهید.

تنظیمات اولیه فایل Layout سایت

پس از دریافت بسته‌ی نیوگت jqGrid، نیاز است فایل‌های مورد نیاز اصلی آن‌را 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 jQuery UI است و نیازی نیست تا سایر فایل‌های آن‌را لحاظ کرد.
این گرید به همراه فایل زبان فارسی grid.locale-fa.js نیز می‌باشد که در کدهای فوق پیوست شده‌است. البته اگر فرصت

کردید 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;
}
```

احتمالا تنظیمات قلم‌های jQuery UI 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 ASP.NET MVC صرفا 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.
Total، نمایانگر تعداد صفحات اطلاعات است. عدد Page، شماره صفحه‌ی جاری است. عدد Records، 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.
UserData اختیاری است. برای مثال اگر خواستید جمع کل صفحه را در ذیل گرید نمایش دهید، می‌توانید یک anonymous object را در اینجا مقدار دهی کنید. خاصیت‌های آن دقیقا 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="rsperror"></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 wich 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: ['قیمت' ,'موجود است' ,'نام محصول' ,'شماره'],
                //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("#rsperror").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.

Div سومی با id مساوی rsperror نیز تعریف شده‌است که از آن جهت نمایش خطاهای بازگشت داده شده از سرور استفاده کرده‌ایم.

- در ادامه نحوه‌ی فراخوانی افزونه‌ی jqGrid را بر روی جدول list ملاحظه می‌کنید.

- خاصیت caption، عنوان نمایش داده شده در بالای گرید را مقدار دهی می‌کند:

آزمایش اول

	شماره	نام محصول	موجود است	قیمت
1	1	نام 1	☑	1000
2	2	نام 2	☐	1001
3	3	نام 3	☑	1002
4	4	نام 4	☐	1003
5	5	نام 5	☑	1004
6	6	نام 6	☐	1005
7	7	نام 7	☑	1006
8	8	نام 8	☐	1007
9	9	نام 9	☑	1008
10	10	نام 10	☐	1009
		جمع صفحه		10045

نمایش 1 - 10 از 500 صفحه 1 از 50 10 تنظیم نمایش ستون‌ها

- خاصیت url، به آدرسی اشاره می‌کند که قرار است ساختار JqGridData ایی را که پیشتر در مورد آن بحث کردیم، با فرمت JSON بازگشت دهد. در اینجا برای مثال به یک اکشن متد 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 دات نت، 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 پیش فرضی 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 JqGridData به RowCells 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 ایی 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 rtl تنظیم شود.
- colModel 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 پیش فرض، ستون جاری رشته‌ای مرتب شود یا اینکه برای مثال عددی پردازش گردد. مقادیر مجاز آن text (مقدار پیش فرض)، date، integer، int، numeric، currency، number، float و datetime

هستند.

- در ستون IsAvailable، مقدار formatter نیز تنظیم شده‌است. در اینجا توسط formatter، 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 گرید ظاهر می‌شود.

- خاصیت sortname، نحوه‌ی مرتب سازی اولیه گرید را مشخص می‌کند.

- خاصیت sortorder، جهت مرتب سازی اولیه‌ی گردید را تنظیم می‌کند.

- viewrecords: تعداد رکوردها را در نوار ابزار پایین گرید 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 گرید، که در اینجا به auto تنظیم شده‌است.

- height: طول گرید، که در اینجا به auto جهت محاسبه‌ی خودکار، تنظیم شده‌است.

- gridview: برای بالا بردن سرعت نمایشی به true تنظیم شده‌است. در این حالت کل ردیف یکباره درج می‌شود. اگر از subgird 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 خطایی صادر شود، در این callback قابل دریافت خواهد بود.

- در ادامه توسط فراخوانی متد 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 متد jqGrid با پارامتر navButtonAdd در ناحیه pager، سبب نمایش دکمه‌ای که صفحه‌ی انتخاب ستون‌ها را ظاهر می‌کند، خواهیم شد.

پیشنیاز 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 dynamic LINQ استفاده کرد.

```
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 متد 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
            {
                UserData = new // نمایش در فوتر
                {
                    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.Id.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 متد GetProducts نیز مهم است. دقیقا همین پارامترها با همین نام‌ها از طرف jqGrid به سرور ارسال می‌شوند 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) متد OrderBy نوشته شده، به صورت پویا عمل می‌کند و از کتابخانه‌ی Dynamic LINQ مایکروسافت بهره می‌برد.

به علاوه توسط Take و Skip کار صفحه بندی و بازگشت تنها بازه‌ای از اطلاعات مورد نیاز، انجام می‌شود.

ب) لیست جنریک محصولات، در نهایت باید با فرمت JqGridData به صورت JSON بازگشت داده شود. نحوه‌ی این Projection را در اینجا می‌توانید 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 می‌توانید 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 ۱۳:۳۳

امکان دارد شکل گرید را بصورت گرید بوت استرپ نمایش داد ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۱۱ ۱۴:۱۰

همانطور که در بحث عنوان شد، قالب این گرید از jQuery UI تامین می‌شود. بنابراین هر نوع قالب نوشته شده‌ای برای jQuery UI با آن سازگار است؛ مانند:

- jQuery UI Bootstrap
- jqGrid.bootstrap

نویسنده: مهدی سعیدی فر
تاریخ: ۱۳۹۳/۰۴/۱۱ ۱۹:۲۱

متاسفانه در حالت rtl تغییر اندازه‌ی ستون‌ها با باگ همراه هست.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۱۱ ۲۰:۶

چه مشکلی دقیقا؟

نویسنده: مهدی سعیدی فر
تاریخ: ۱۳۹۳/۰۴/۱۱ ۲۰:۳۰

مثلا در اینجا سایز ستون Id را بیشتر کردم، ردیف‌های متناظرش باهاش هماهنگ نیستند.(با موس اندازه‌ی ستون را تغییر دهیم)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۱۱ ۲۰:۵۱

درسته. البته فقط با کروم 35 اینطوری هست. با IE 11 و فایرفاکس 30 تست کردم مشکلی نبود.

نویسنده: مهدی سعیدی فر
تاریخ: ۱۳۹۳/۰۴/۱۱ ۲۳:۴۳

خیلی عالی. خیلی وقته که دنبال یک دیتا گرید می‌گردم تا مشکلی نداشته باشه. برای AngularJS یک Datagrid هست به نام ng-grid که خیلی خوب و کامل هست ولی rtl را خوب ساپورت نمی‌کنه. نسخه‌ی rtl هم که یه نفر براش 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 می‌خوام از ng-grid کوچ کنم به jq-grid و ببینم ارزش این مهاجرت را داره یا نه.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۱۲ ۰:۱۹

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: ۱۳۹۳/۰۴/۱۶ ۲۲:۱۳

وقتی این کتاب خانه (Install-Package 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: ۱۳۹۳/۰۴/۱۶ ۲۲:۴۵

این بسته، کامپایل شده‌ی آن فایل را تحت عنوان Dynamic.dll به لیست ارجاعات پروژه اضافه می‌کند. فایل Dynamic Expressions.html هم راهنمای آن است. به عبارتی همه چیز آماده است؛ از فضای نام جدید System.Linq.Dynamic آن استفاده کنید.

نویسنده: daniyal
تاریخ: ۱۳۹۳/۰۴/۲۱ ۱۵:۲۷

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

تاریخ: ۱۳۹۳/۰۴/۲۱ ۱۸:۵۶

columnChooser رویداد done هم دارد که از آن می‌شود برای آنالیز ستون‌ها استفاده کرد:

```
.jqGrid('navButtonAdd', '#pager',
            {
                caption: "تنظیم نمایش ستون‌ها", title: "Reorder Columns",
                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 }
                            });
                        }
                    });
                }
            })
```

متد getGridParam با پارامتر colModel، آرایه‌ای از خواص ستون‌ها را بر می‌گرداند.

```
var colModel = $("#list").jqGrid('getGridParam', 'colModel');
```

در این خواص اگر hidden مساوی true بود، یعنی مخفی شده‌است. 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 متدی است که لیست نام ستون‌های مخفی را دریافت می‌کند.

نویسنده: daniyal
تاریخ: ۱۳۹۳/۰۴/۲۲ ۹:۳۰

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 بخوام شفاف‌تر بگم ، من می‌خوام وقتی کاربر دکمه خروجی پی دی اف رو زد ستون‌های گرید رو کاربر قبلش تنظیم کرده باشه و در پی دی اف مورد نظر ستون‌های حذف شده رو دیگر نبینه همچنین با در نظر گرفتن موارد فیلتر شده در گرید .

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۲۲ ۱۲:۲

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 در تعریف ستون‌ها (مثلا ستون Id):

```
column.IsVisible(hiddenColumns.Split(',').All(col => col != "Id"));
```

یک نکته: ذکر function در postData ضروری است؛ وگرنه فقط یکبار محاسبه می‌شود.

نویسنده: alireza
تاریخ: ۱۳۹۳/۰۵/۱۰ ۱۵:۳۲

سلام خسته نباشید. 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: ۱۳۹۳/۰۵/۱۰ ۱۷:۳۳

- 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 MVC 5 آن به صورت صحیح به پروژه اضافه می‌شوند.

نویسنده: alireza
تاریخ: ۱۳۹۳/۰۵/۱۱ ۱۱:۳

با mvc5 باید باشه ؟ من mvc 4 دارم

نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۰۵/۱۱ ۱۱:۲۱

خیر. یک پروژه خالی MVC4 ایجاد کنید. بعد پوشه‌ها و فایل‌های این پروژه را در آن اضافه کنید (منهای فایل‌های کانفیگ)؛ کار می‌کند. 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 شده‌است. در این مثال خاص، کاربرد ویژه‌ی دیگری ندارد. می‌شد بجای آن از Web API هم استفاده کرد.

نویسنده: ناصر پورعلی
تاریخ: ۱۳۹۳/۰۵/۱۴ ۱۴:۲۲

سلام؛ از مثالی که زدید دارم استفاده میکنم، در کلاس ReflectionHelper متد FindFieldType موقعی که از جداول خود ارجاع دهنده استفاده میکنیم(در اینجا جدول پست)، خطا stackoverflow می‌گیرم، کجاش رو باید چک کنیم که فراخوانی بی نهایت اتفاق نیفته؟ مگه فیلد dumplevel همچین کاری انجام نمیده؟

data model م به این صورته :

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۱۴ ۱۵:۶

«مگه فیلد dumplevel همچین کاری انجام نمیده؟»

بله. dump level از stack overflow جلوگیری می‌کند. اما جای دیگری است که stack overflow واقعی رخ می‌دهد:

« بررسی خطای Circular References در ASP.NET MVC Json Serialization »

نویسنده: بهاره
تاریخ: ۱۳۹۳/۰۵/۲۰ ۱۶:۰

سلام

اگه بخوام نام ستون‌ها رو از روی مدل یا ویو بگ بخونم و دستی ننویسم باید چیکار کنم. من از مدل Database First استفاده

می‌کنم. چطور میتونم اینکار رو انجام بدم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۲۰ ۱۷:۱۴

« استفاده ازExpressionها جهت ایجاد Strongly typed view در ASP.NET MVC »

نویسنده: احسان شیروان
تاریخ: ۱۳۹۳/۰۶/۱۲ ۲۱:۷

با سلام و خسته نباشید من یه مشکلی داشتم می‌خواستم در صورت امکان راهنمایی نمایید :
توی یه صفحه می‌خوام لود شدن داده‌های این گرید با رویداد کلیک یه دکمه اتفاق بیفته یه مقدار سرچ هم زدم که تقریبا همشون می‌گفتن توی رویداد کلیک بنویسم :

```
jQuery("#list").trigger("reloadGrid");
```

اما با این هم جواب نداد و اصلا ActionMethod فراخوانی نمیشه ممنون میشم راهنمایی نمایید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۶/۱۲ ۲۱:۲۴

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 متد جداگانه قرار دهید و سپس آن‌را مستقلا فراخوانی کنید.
document.ready یعنی به محض آماده شدن DOM این اطلاعات را اجرا کن.

نویسنده: احسان شیروان
تاریخ: ۱۳۹۳/۰۶/۱۲ ۲۱:۳۹

راستش اولش هم همین کارو کردم جواب نگرفتم تستش هم کردم مطمئنم که رویداد کلیک و تابع مذکور فراخوانی میشه اما متد سرور فراخوانی نمیشه

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۶/۱۲ ۲۲:۳۸

پس از آزمایش مشکلی مشاهده نشد.
همانطور که عنوان شد، کل اطلاعات داخل document ready به داخل یک متد مجزا 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 ۱۳:۹

یک نکته‌ی تکمیلی

نسخه‌ی جدید و زنده‌ی dynamic LINQ در اینجا نگهداری می‌شود.

```
PM> Install-Package System.Linq.Dynamic.Library
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۰۲ ۱۴:۲۱

برای رفع این مشکل فقط کافی است .ui-jqgrid-hbox-rtl. ui-jqgrid را در فایل css آن یافته و تبدیل کنید به:

```
.ui-jqgrid .ui-jqgrid-hbox-rtl {float: right; padding-left: 0;}
```

نویسنده: سهراب
تاریخ: ۱۳۹۳/۰۸/۱۷ ۱۸:۱۷

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: ۱۳۹۳/۰۸/۱۷ ۱۹:۲۵

مجوز عمومی فایل‌های اسکریپت اصلی آن MIT است و در هر نوع پروژه‌ای قابل استفاده‌است. مجوز تجاری هم دارد برای حالتیکه بخواهید کامپونت‌های ASP.NET آن‌را بخرید 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: ۱۳۹۳/۰۹/۰۸ ۱۳:۱۰

با تشکر از آقای نصیری

میخواستم بدونم jqgrid امکان دیتا بایندینگ داره به صورتی که وقتی یه رکورد اضافه کردیم و اونا 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: ۱۳۹۳/۰۹/۰۸ ۱۳:۴۷

مطالب 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 لابلای مباحث این گروه با ارائه کدهای مرتبط، بحث شده‌است.

نویسنده: حامد شاه محمدی
تاریخ: ۱۳۹۳/۰۹/۱۱ ۱۶:۲۰

من میخواستم jqgrid رو به بانک اطلاعاتی وصل کنم ولی اسکریپت jquery اطلاعات رو برای اکشن متدم نمیفرسته break point گذاشتم ولی اتفاقی نمیفته مشکل از چیه؟

نویسنده: وحید نصیری

تاریخ: ۱۳۹۳/۰۹/۱۱ ۱۷:۲۴

- مثال کامل بحث در انتهای آن پیوست شده‌است. کار خودتان را با آن مطابقت دهید.
- همچنین نیاز است با مباحث دیباگ اسکریپت‌ها آشنا باشید. اطلاعات بیشتر
+ آشنایی با نحوه‌ی صحیح طرح یک سؤال فنی هم در تمام سایت‌ها راهگشا است. اطلاعات بیشتر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۱۹ ۱۴:۴۸

جهت اطلاع

« تغییر نام و همچنین مجوز استفاده از jqGrid »
- به عبارتی تا نگارش **4.6** آن رایگان است و نه پس از آن.
- به همین جهت از نگارش 4.6 آن یک fork جدید ایجاد شده در اینجا

نویسنده: زهره ق
تاریخ: ۱۳۹۳/۱۲/۱۷ ۱۵:۱۷

باسلام؛ من در قسمت نحوه ارتباط پایگاه داده با jqgrid مشکل دارم. در asp میامدیم DataSource گرید را مساوی با DataTable قرار میدادیم ولی الان نمیدونم چه جوری داده‌های داخل پایگاه داده رو داخل jqGrid قرار بدم یا داده جدیدی از طریق jqgrid به پایگاه داده اضافه کنم. داده‌های من که حاصل اجرای storeProcedure است. داخل یک datatable قرار داره چه جوری میتونم jqgrid رو با اون ارتباط بدهم؟ تا داده‌های من رو نشون بده. ممنون میشم راهنماییم کنید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۲/۱۷ ۱۵:۴۴

- یک سری micro orm مانند Dapper وجود دارند که اساس کار آن‌ها، کوئری نویسی و سپس دریافت اطلاعات از طریق لیست‌های جنریک (مانند روش نهایی کار در مثال جاری) است. بنابراین datatable را فراموش کنید و شروع کنید به استفاده از dapper یا peta poco (اگر نمی‌خواهید از Entity framework استفاده کنید).
- مهم این نیست که منبع داده‌ی شما به چه نحوی تهیه می‌شود. مهم این است که این گرید خاص و امثال آن (Kendo UI هم به همین صورت)، داده‌ها را با فرمت JSON دریافت می‌کنند. همچنین داده‌ها را با فرمت JSON به سمت سرور ارسال می‌کنند. بنابراین مطابق مثالی که زده شد، اطلاعات خودتان را تبدیل می‌کنید به فرمت new JqGridData و سپس آن‌را به صورت JSON بازگشت می‌دهید. همچنین در سمت کاربر، تمام تغییرات، با فرمت JSON به سمت سرور ارسال می‌شوند که این مورد و آنالیز آن در قسمت‌های بعدی این سری بررسی شده‌اند.
- در این سری مثال‌ها از امکانات توکار ASP.NET MVC برای بازگشت داده‌ها به صورت JSON استفاده شده‌است. اما jqGrid یا Kendo UI Grid و امثال آن‌ها اساسا هیچگونه وابستگی به فناوری‌های سمت سرور ندارند. بنابراین از یکی از روش‌های موجود بازگشت اطلاعات به فرمت JSON استفاده کنید. مهم هم نیست که برنامه‌ی شما ASP.NET MVC است یا وب فرم یا PHP یا هر چی. برای مثال از کتابخانه‌ی JSON.NET برای بازگشت اطلاعات به فرمت JSON استفاده کنید.
- علاوه بر این، روش‌های دیگری هم برای بازگشت اطلاعات به فرمت JSON وجود دارند. اطلاعات بیشتر

نویسنده: زهره ق
تاریخ: ۱۳۹۳/۱۲/۱۸ ۱۶:۱۰

با سلام؛ خیلی ممنونم. ولی میخواستم اون کاری که میخوام انجام بدم رو بیشتر توضیح بدم تا موضوع روشن‌تر بشه

همون طور که در عکس مشخصه من 3 تا جدول دارم

-Programs برای نگه داری اطلاعات برنامه که به صورت تو در تو است

-PrgElements برای نگه داری اطلاعات مربوط به المان‌های تعریف شده در برنامه (مثل فرم -منو- کنترل و ...) که ElementID به صورت HiearachyID تعریف شده و متاسفانه Entity Framwork,LingToSql این DataType رو ساپرت نمیکنه برای همین باید از ADO استفاده کنم.

من میخوام گریدی داشته باشم که کاربر اطلاعات راجع به المنت‌های برنامه رو تو در تو وارد کنه و این اطلاعات برای کاربر نمایش داده بشه تا کاربر بتونه با انتخاب سطر مربوطه اون رو ویرایش کنه حذف کنه و مورد جدیدی رو اضافه کنه

یک گریدی به این شکل:

من در قسمت کدهای سمت سرور ، اون قسمتی که row رو مقدار دهی میکنید رو متاسفانه نمیتونم پیاده سازی کنم (در واقع تبدیل داده مورد نظر به فرمت jqGridData جایی که Row رو مقدار دهی می‌کنیم)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۲/۱۸ ۱۶:۴۴

مطلب اختصاصی و همچنین نظرات بحث « نمایش ساختارهای درختی توسط jqGrid » را مطالعه کنید.

عنوان: **فرمت کردن اطلاعات نمایش داده شده به کمک jqGrid در ASP.NET MVC**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۱۲ ۸:۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, jqGrid

پیشنیاز 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 نمایش داده شود.
ج) عدد قیمت با سه رقم جدا کننده همراه باشد.

تکمیل مدل برنامه

مدل قسمت اول صرفا یک محصول بود. مدل قسمت جاری، اطلاعات تولید/تامین کننده آن‌را توسط کلاس Supplier نیز به همراه دارد:

```
namespace jqGrid02.Models
{
    public class Product
    {
        public int Id { set; get; }
        public string Name { set; get; }
```

```
        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 دقیقا مانند قسمت قبل است.

در اینجا فقط یک اکشن متد جدید جهت بازگشت اطلاعات تولید کننده‌ای مشخص با فرمت JSON، اضافه شده‌است:

```
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 گرید 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'
            });

            $('#list').jqGrid({
                // .... مانند قبل
                colNames: ['قیمت' ,'نام محصول' ,'شماره'],
                //columns model
                colModel: [
                    {
                        name: 'Id', index: 'Id', align: 'right', width: 20,
                        formatter: function (cellvalue, options, rowObject) {
                            var cellValueInt = parseInt(cellvalue);
                            if (cellValueInt == 5) {
                                return "<span style='background: brown; color: yellow'>" + cellvalue +
"</span>";
                            }
                            return cellvalue;
                        }
                    },
                    {
                        name: 'Name', index: 'Name', align: 'right', width: 300,
                        formatter: function (cellvalue, options, rowObject) {
                            return "<a href='#' onclick='showSupplierDialog(this, " + rowObject[0] +
");'>" + cellvalue + "</a>";
                        }
                    },
                    {
                        name: 'Price', index: 'Price', align: 'center', width: 50,
                        formatter: 'currency',
                        formatoptions:
                        {
                            decimalSeparator: '.', thousandsSeparator: ',', decimalPlaces: 2, prefix:
'$'
                        }
                    }
                ],
                // .... مانند قبل
            });
        });
    </script>
```

- همانطور که ملاحظه می‌کنید، توسط خاصیت formatter می‌توان عناصر در حال نمایش را فرمت کرد و بر روی نحوه‌ی نمایش نهایی آن‌ها تاثیرگذار بود.

در حالت ستون Id، از یک formatter سفارشی استفاده شده‌است. 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 خواهند 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 شده‌است. نمونه‌ای از آن را در قسمت اول در ستون نمایش وضعیت موجود بودن محصول با تنظیم formatter: checkbox مشاهده کرده‌اید. در اینجا از یک formatter توکار دیگر به نام currency برای کار با مقادیر پولی استفاده شده‌است به همراه تنظیمات خاص آن.

- متد showSupplierDialog طوری تنظیم شده‌است که پس از دریافت Id یک محصول، آن‌را به سرور ارسال کرده و مشخصات تولید کننده‌ی آن‌را با فرمت JSON دریافت می‌کند. سپس در حلقه‌ای که مشاهده می‌کنید، خواص شیء جاوا اسکریپتی دریافتی استخراج و به spanهای supplierDialog انتساب داده می‌شوند. جهت سهولت کار، Id این spanها دقیقا مساوی Id خواص شیء دریافتی از سرور، درنظر گرفته شده‌اند.

- در مورد راست به چپ نمایش داده شدن عنوان دیالوگ، تغییرات CSS ایی لازم است که در قسمت اول بیان شدند.

برای مطالعه بیشتر

لیست کامل فرمت کننده‌های توکار

فرمت کننده‌های سفارشی

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید

jqGrid02.zip

عنوان: فعال سازی و پردازش جستجوی پویای jqGrid در ASP.NET MVC
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۱۳ ۸:۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, jqGrid

پیشنیاز 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، ابتدا 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، جهت دو ستون نام محصول و نام تولید کننده، تنظیم شده‌اند.
- stype، روش مقایسه‌ی مقادیر را مشخص می‌کند. مقدار پیش فرض آن text است و مقادیری مانند int، integer، float، number، numeric، 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'];
```

eq 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.
- searchoptions گزینه‌های دیگری را نیز می‌تواند شامل شود. برای مثال در حین نمایش جستجوی داخل ردیفی یا صفحه‌ی 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 شده‌است.
SuppliersSelect آن به اکشن متد ذیل اشاره می‌کند که لیست تولید کنندگان را با فرمت لیستی از SelectListItemها 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 dorp down پویا نمایش داده خواهد شد.

- اگر دقت کرده باشید، نام فیلد تولید کننده با Supplier.Id مقدار دهی شده‌است. 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.

- 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 می‌شود 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 navGrid، در ناحیه‌ی pager گرید، تنظیمات جستجو را فعال کرده‌ایم.

multipleSearch 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، مجددا از متد jqGrid با پارامتر navButtonAdd در ناحیه‌ی pager استفاده خواهیم کرد. کلیک بر روی آن سبب اجرای متد

toolbarSearching می‌شود.

در اینجا حداقل سه نوع جستجو را می‌توان فعال کرد:
- filterToolbar که سبب نمایش نوار ابزار جستجو، دقیقا 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 هربار تنها بر روی یک فیلد قابل انجام خواهد بود و برعکس.
- در حالت جستجوی نوار ابزاری، اگر خواص searchOperators و stringResult به true تنظیم شوند، 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 مثلا contains یا تساوی و امثال آن و نه حالت پیشرفته‌ی انتخاب عملگرها توسط کاربر.

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&search
Oper=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%22fie
ld%22%3A%22SupplierID%22%2C%22op%22%3A%22eq%22%2C%22data%22%3A%221%22%7D%5D%7D&searchField=&searchStrin
g=&searchOper=
// filters ->
{"groupOp":"AND","rules":[{"field":"All","op":"cn","data":"fffff"},{"field":"Price","op":"bn","data":"f
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 stringResult به true تنظیم نشوند، حالت toolbar search فوق را شاهد خواهیم بود. در غیراینصورت به حالت multi-field search سوئیچ می‌شود.

- 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 متدی 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 متد 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 متد اضافه شده‌اند، تا فیلدهای جستجو را نیز دریافت کنند.
- نوع متد به HttpPost تغییر کرده‌است. این مورد برای ارسال اطلاعات حجیم 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: ۱۳۹۳/۰۴/۲۷ ۱۵:۲

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: ۱۳۹۳/۰۴/۲۷ ۱۵:۲۴

با استفاده از متد toggleToolbar می‌شود نوار ابزار نمایش داده شده را مخفی یا مجددا نمایش داد:

```
<button onclick="$('#list')[0].toggleToolbar()">نمایش/حذف تولبار</button>
```

نویسنده: شهربانو مشهدی
تاریخ: ۱۳۹۳/۰۴/۲۷ ۱۷:۴۴

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: ۱۳۹۳/۰۴/۲۷ ۱۷:۵۳

- در کدهای شما ابتدا نوار ابزار نمایش داده می‌شود، سپس toggle تا خاموش شود.
+ این‌ها یک سری مثال هستند برای نمایش نحوه‌ی 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 «زنجیر 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: ۱۳۹۳/۰۴/۲۷ ۲۰:۱۰

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: ۱۳۹۳/۰۵/۱۱ ۲:۳۷

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: ۱۳۹۳/۰۵/۱۱ ۹:۳۴

« پیدا کردن آیتم‌های تکراری در یک لیست به کمک LINQ »

عنوان: **استفاده از چند Routing در یک پروژه ASP.NET MVC بدون درد و خونریزی**
نویسنده: مرتضی دلیل
تاریخ: ۱۳۹۳/۰۴/۱۳ ۱۰:۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, routing

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 روبرو 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 اصولا 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، متد 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 پُر می‌شود. بخشی از فایل RoutConfig.cs را در تصویر زیر می‌بینیم.

```
1 reference
public class RouteConfig
{
    1 reference
    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 }
        );
```

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 متد (اکشن) 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 MapRouting فوق متفاوت است.**

اما این اسم‌ها(name 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:

```
<a href="@Url.Action("post", "News", new { id = @item.PostName }, null)"> @item.PostTitle </a>
```

فرض کنید یک سایت خبری داریم و گروه‌های مختلف آن شامل مطالب متفاوتی هستند. کنترلری به نام News ایجاد کرده‌ایم، که دارای یک اکشن به نام post است که نام مطلب را دریافت کرده و یک View به ما برمی‌گرداند. 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 Url.Action ساخته می‌شود. ما یک MapRouting به نام Default در کالکشن Routes داریم، پس Url.Action لینک زیر را می‌سازد:

```
http://mydomain.com/news/post/نام-مقاله
```

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 یا https 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):

```
<a href="/News">اخبار</a>
```

واضح است که چون MapRouting یک حالت پیش فرض درونی دارد که دارای اکشن دیفالت index است، پس ActionLink 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 آبجکتِ 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. MapRouting فعلی جوابگوی ما نخواهد بود و آدرس را به شکل زیر نمایش میدهد.

```
<a href="/News/index?pid=2">صفحه بعد</a>
```

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 take و 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 بخش هایی از برنامه نیاز به روتینگ پیدا کنید و روتینگ‌های جدید ایجاد کنید متوجه خواهید شد که مدیریت این MapRouting ها کار خسته کننده و طاقت فرسایی خواهد شد، مخصوصا اگر بدانید که فقط مجاز به استفاده از یک پارامتر optional در هر MapRouting هستید! دست شما کاملا بسته است. لینک‌های بالای سایت را اصلاح میکنید ولی لینک‌های پایین سایت خراب میشوند و بالعکس.

به هر حال MapRouting زیر را به RoutConfig.cs اضافه میکنیم :

```
routes.MapRoute("PostPaging", "{controller}/{action}/{id}/{pid}",
            defaults: new
            {
                controller = "News",
                action = "Index",
                id = "all",
                pid = UrlParameter.Optional

            }
            );
```

- اسم این MapRouting ، دیگر Default 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 درلینکهایمان نبرده باشیم به شکل پیشفرض all قرار داده میشود که یعنی کل اخبار مد نظر است. (در اکشن مربوطه باید این تصمیمات را لحاظ کنیم)

- حتما این MapRouting را بعد از MapRouting اولیه بنویسید، کمی پیشتر، علت این امر توضیح داده شد و گفته شد اولین چیزی که MVC پس از درخواست ما میبیند به عنوان Routing بررسی می‌کند (درخواست اولیه) و چون ساختار MapRouting فوق تا اندازه ای 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 حتا اگر قبل از مپ روتینگ دیفالت نوشته شود برنامه با مشکل مواجه نخواهد شد، چون اصلا شکل درخواست اولیه به سایت، چیزی شبیه این آدرس نیست. اما خاص بودن آن و همچنین نوع بهره گیری از آن با کمک Action یا ActioLink 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("اخبار", "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 :

```
<a href="/News/index?pid=2">صفحه بعد</a>
```

علت واضح است، ActionLink اصلا MapRouting 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 و ActionLink های 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 کاملا به ترتیب ثبت مپ روتینگ‌ها در کالکشن Routes بستگی دارد .

خوشبختانه 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 :

```
<a href="/News/cat/all/2">صفحه بعد</a>
```

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 MapRouting با نام default بازنویسی میکنیم و نتیجه چیزی شبیه کد زیر خواهد شد :

```
<a href="/News">اخبار</a>
```

در RoutLink 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 پراپرتی‌ها 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 MapRoutnig را قبل از دیگر Routing ها می‌نویسیم؟ وسط دو MapRouting قبلی مینویسیم؟ آخر MapRouting ها مینویسیم؟ آیا فرقی میکند؟ اگر سریع بگوییم خـــیر! اشتباه کرده ایم. واقعا 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 اصلا ربطی به نامی که به RoutLink داده ایم ندارد و ترتیب موجود در کالکشن ایجاد شده **در RoutConfig تعیین کننده است.(آبجکت Routes) اگر MapRouting فوق را در انتهای بقیه بگذاریم صفحه اول لود میشود ولی با کلیک روی "درباره ما" صفحه پیغام خطا خواهد داد.**

باید به یاد داشته باشیم برای اجرای درخواست (کلیک روی لینک)، آنچه برای ASP.NET MVC اهمیت دارد، ترتیب قرار گیری MapRouting ها در RouteRegister 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 اصلا 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 اصلا چیزی به نام page در آدرس وجود ندارد که از این MapRouting بخواهد پیروی کند. اما در شماره 2 گیر می‌افتد چون این فرمت را حفظ کرده است :

```
"{controller}/{action}/{id}/{pid}"
```

Pid که اختیاریست (بر اساس قوانین تعریف شده در مپ روتینگ شماره 2) بقیه موارد نیز حالت دیفالت دارند، یعنی اگر تعریف نشده باشند (مثل همین درخواست) خودبخود جایگذاری میشوند. پس به طور کلی صفحه اول ما تغییر می‌کند و اکشن index 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. اولا 1 قبل از 2 است و صفحه اول برای لود شدن به مشکل برخورد نمیکند.
ثانیا وقتی روی "درباره ما" کلیک میکنیم همان شماره 3 فراخوانی میشود و بقیه مپ روتینگ‌ها اعمال نمیشوند.

تنظیم این مقاله با هدف آموزش صورت گرفته است. حتما 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

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۴/۱۹ ۱۶:۴۲

شبیه به مطلب مخفی کردن کوئری استرینگ‌ها در ASP.NET MVC توسط امکانات Routing هست.

عنوان: **فعال سازی و پردازش صفحات پویای افزودن، ویرایش و حذف رکوردهای jqGrid در ASP.NET MVC**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۱۴ ۸:۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, jqGrid

پیشنیاز 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:

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:

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,
                        resize: false, formatter: 'actions',
                        formatoptions: {
                            keys: true
                        }
                    }
                ],
```

- در اینجا هر ستونی که دارای خاصیت editable مساوی true است، قابلیت ویرایش پیدا می‌کند.
- edittype 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 editoptions و سپس dataUrl آن‌را 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 متد SuppliersSelect اشاره می‌کند که نهایتا 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 text، textarea، select، checkbox، password، button، image، file و custom هستند.
- خاصیت editrules، برای مباحث اعتبارسنجی اطلاعات ورودی توسط کاربر پیش بینی شده‌است. 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 minValue و maxValue از نوع عددی، email, url, date, time از نوع bool و custom قابل تنظیم است (مثال‌های حالت 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 چهارم",
   //url from wich data should be requested
   url: '@Url.Action("GetProducts","Home")',
   //url for edit operation
   editurl: '@Url.Action("EditProduct","Home")',
```

اکشن متد 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 متد 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 گرید است. اگر بخواهیم آن‌ها را دستی فعال کنیم (جهت اطلاعات عمومی) می‌توان از فراخوانی متد ذیل نیز کمک گرفت:

```
var lastSel;
        function inlineEdit() {
            $('input[name=rdEditApproach]').attr('disabled', true);
            $('#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 شده‌است. سپس با فراخوانی دستی متد editRow در زمان کلیک بر روی یک ردیف، همان کاری را که ستون actions در جهت فعال سازی خودکار حالت ویرایش سلول‌ها انجام می‌دهد، می‌توان شبیه سازی کرد. متد saveRow نیز کار ارسال اطلاعات تغییر کرده را به سرور انجام می‌دهد.

- برای فعال سازی خودکار فرم‌های افزودن رکوردها و یا ویرایش ردیف‌های موجود می‌توان از فراخوانی متد formEdit ذیل کمک گرفت:

```
function formEdit() {
            $('input[name=rdEditApproach]').attr('disabled', true);
            $('#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";
        }
```

ابتدای 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:

با کلیک بر روی دکمه‌ی افزودن ردیف جدید، صفحه‌ی ذیل به صورت خودکار تولید می‌شود:

و با کلیک بر روی دکمه‌ی ویرایش ردیفی انتخاب شده، صفحه‌ی ویرایش آن ردیف به همراه مقادیر سلول‌های آن ظاهر خواهند شد:

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 شده‌است. توسط checkOnUpdate: true, checkOnSubmit: true سبب خواهیم شد تا اگر کاربر مقادیر موجود فرمی را تغییر داده‌است 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 کرده‌ایم. اگر به ستون پس از شماره‌های خودکار ردیف‌ها، در سمت راست گرید دقت کنید، یک سری chekbox 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 selarrrow 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 بندی 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 آن‌را 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 بهتر است. مجوز عمومی آن MIT است و در هر نوع پروژه‌ای قابل استفاده‌است. مجوز تجاری هم دارد برای حالتیکه بخواهید کامپوننت‌های ASP.NET آن‌را بخرید 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 نباشید.
اگه بخواهیم توی ویرایش به صورت Dialog به جای DropDownList از 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 میخوام تنظیم کنم. رشته html رو تولید میکنم و پاس میدم به خروجی، اما خطا 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>";
```

```
            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
<عملکردی/><option value='2' option><بودجه ای/option>

چه نکته ای وجود داره؟ ایا مربوط به webApi هست؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۲۰ ۱۴:۲۱

- زمانیکه از یک اکشن متد، خروجی HTML دریافت می‌کنید، Content-Type آن مساوی text/html 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 ایی دارد).
- ضمنا 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 شده‌است:

```
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 EmployeeId و EmployeeName ها). در رویدادگردان سمت کاربر buildSelect ، این مورد دریافت و پردازش می‌شود.

نویسنده: ابوالفضل رجب پور
تاریخ: ۱۳۹۳/۰۵/۲۱ ۱۵:۱۵

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: ۱۳۹۳/۰۵/۲۱ ۱۵:۳۴

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: ۱۳۹۳/۰۹/۰۶ ۱۳:۱۱

با سلام؛ من میخوام در هنگام Add, Update , Delete هنگام ارسال اطلاعات به سرور ValidateAntiForgeryToken رو هم بفرستم. چطور میشه اینکارو انجام بدم، روش‌های مختلفی رو جستجو و امتحان کردم ولی جواب نداد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۰۶ ۱۳:۳۸

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: ۱۳۹۳/۰۹/۰۶ ۱۶:۵۹

property خاصی از jqgrid باید مقدار دهی بشه؟ متاسفانه جواب نداد!

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۰۶ ۱۷:۴۲

@Html.AntiForgeryToken() را روی صفحه قرار دادید؟ بررسی name=__RequestVerificationToken در کد فوق، بر مبنای وجود این فیلد در صفحه‌است.

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 متدهای ثبت، ویرایش و حذف، ویژگی ValidateAntiForgeryToken اضافه شد.
تصویر فوق هم بر همین مبنا تهیه شده‌است.
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 گرید 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 شده‌است. به علاوه نحوه‌ی استفاده از anti-forgery token در آن ذکر شده، به همراه StronglyTyped.PropertyName ها.

نویسنده: امیر نوروزیان
تاریخ: ۱۳۹۳/۱۰/۰۹ ۲۱:۱۸

ممنون. 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: ۱۳۹۳/۱۰/۰۹ ۲۲:۳

مثال به روز شده را اجرا کردید؟ شماره Id اولین ردیف را به عمد اینبار از 100 تنظیم کردم (تا با شماره ردیف اشتباه 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):

```
55      [ValidateAntiForgeryToken]
56      [HttpPost]
57      public ActionResult DeleteProduct(string id)
58      {                                          id  "100"
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      }
```

عنوان: **رمزنگاری خودکار فیلدهای مخفی در ASP.NET MVC**
نویسنده: محسن موسوی
تاریخ: ۱۳۹۳/۰۴/۱۴ ۲۲:۴۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, Security

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 رمز گشایی آن هنگام Post شدن سمت سرور می‌باشد.

برای رسیدن به این هدف یک Understanding and Extending Controller Factory in MVC (Controller Factory) سفارشی را جهت دستیابی به مقادیر فرم ارسالی، قبل از استفاده در Actionها 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.

EncryptionKey : کلید رمز نگاری میباشد و در فایل Config برنامه ذخیره میباشد.

EncryptionPrefix : پیشوند نام Hidden فیلدها میباشد، این پیشوند برای یافتن Hidden فیلد هایی که رمزنگاری شده اند استفاده میشود. میتوان این فیلد را در فایل Config برنامه ذخیره کرد.

```
<appSettings>
    <add key="EncryptionKey" value="asdjahsdkhaksj dkashdkhak sdhkahsdkha kjsdhkasd"/>
  </appSettings>
```

کلاس RijndaelStringEncrypter :

```
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 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.

کلاس ActionKeyService :

```
/// <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 متد 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.

کلاس MvcHtmlHelperExtentions :

```
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 متد در DefaultControllerFactory جهت بدست آوردن کلید View در زمانیکه میخواهیم اطلاعات را بازیابی کنیم استفاده میشود.

```
public static string GetActionKey(this HtmlHelper helper)
```

از این متد در متدهای کمکی درنظر گرفته جهت ایجاد فیلدهای مخفی رمز نگاری شده، استفاده میکنیم.

کلاس InputExtensions :

```
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 متد الحاقی را در Viewهای برنامه، مشاهده مینمایید.

```
@Html.EncryptedHiddenFor(model => model.Id)
   @Html.EncryptedHidden("Id2","2")
```

کلاس DecryptingControllerFactory :

```
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 DefaultControllerFactory 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 UrlParameterها یافت شود و درصورت 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 کاربر یا AntyForgeryToken تولیدی در View را نیز در کلید دخالت داد و در هربار Post شدن اطلاعات این ActionKeyValue مربوط به کاربر جاری را تغییر داد و کلیدها را در بانکهای اطلاعاتی ذخیره نمود.**

مراجع:

Automatic Encryption of Secure Form Field Data

Encrypted Hidden Redux : Let's Get Salty

عنوان: استفاده ازExpressionها جهت ایجاد Strongly typed view در ASP.NET MVC
نویسنده: محسن موسوی
تاریخ: ۱۳۹۳/۰۴/۱۵ ۰:۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, Reflection, Kendo UI, jqGrid

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>
</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,
```

```
                    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 متغییر هایمان ، اطلاعات گرید 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 متغییر هایمان استفاده مینماییم.

```
public static string PropertyDisplay<T>(this Expression<Func<T, object>> expression)
```

جهت بدست آوردن DisplayNameAttribute استفده میشود. درصورتیکه این DisplayNameAttribute یافت نشود نام متغییر بازگشت داده میشود.

بنابراین View مربوطه را اینگونه بازنویسی میکنیم:

```
@using Models
@{
    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>
</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,
                    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
                }]
            });
        });
    </script>
}
```

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 متد 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 GetNestedPropertyPath(Expression 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 GetNestedPropertyPath(expression);
        }
    }
}
```

در این حالت خواهیم داشت:

```
var name = new PropertyHelper().GetNestedPropertyName<CompanyModel>(x => x.Product.Id);
```

که خروجی Product.Id را بر می‌گرداند.

نویسنده: محسن موسوی
تاریخ: ۱۳۹۳/۰۷/۲۶ ۱۸:۸

در نهایت این متد 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.

عنوان: سفارشی سازی عناصر صفحات پویای افزودن و ویرایش رکوردهای jqGrid در ASP.NET MVC
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۱۵ ۸:۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, jqGrid

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، صرفا 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 حین تکمیل نام، یک auto complete ظاهر شود:

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 نشوند.

پیشنیازها

- 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 « PersianDatePicker یک DatePicker شمسی به زبان JavaScript 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.

تعریف و الحاق این پیشنیازها، 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>
</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 شده‌است که جزئی از jQuery UI استاندارد است.

برای پردازش سمت سرور آن و مقدار دهی url آن، یک چنین اکشن متدی 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
                        }
```

```
                }
            ],
```

Date picker مورد استفاده ، وابستگی خاصی به jQuery ندارد. مطابق مستندات آن باید در رویدادگردان onclick، این تقویم شمسی را فعال کرد. بنابراین در قسمت onclick دقیقا این مورد را اعمال می‌کنیم.

```
@{
ViewBag.Title = "Index";
var today = DateTime.Now.ToPersianDate();
}
```

مقدار today آن در ابتدای View به نحو فوق تعریف شده‌است. 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://jquerypriceformat.com/
                            $(elem).priceFormat({
                                prefix: '',
                                thousandsSeparator: ',',
                                clearPrefix: true,
                                centsSeparator: '',
                                centsLimit: 0
                            });
                        }
                    },
                    editrules: {
                        required: true,
                        minValue: 0
                    }
                }
            ],
```

افزونه‌ی price format نیز یک افزونه‌ی جی‌کوئری است. بنابراین دقیقا 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 دقیقا به فرم‌های جستجو نیز می‌توان اعمال کرد. در اینجا برای حالات ویرایش و افزودن رکوردها، editoptions مقدار دهی شده‌است؛ در مورد فرم‌های جستجو باید searchoptions و برای مثال dataInit 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 decaimal 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 مثلا برای Redirect کردن به صفحه دیگه اضافه کنیم باید به چه صورت عمل کنیم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۱۸ ۱۱:۵۴

- از formatterها برای سفارشی سازی و تغییر اطلاعات نهایی نمایش داده شده در یک سلول استفاده کنید.
- از callback ایی به نام 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 () {
        $('<a href="#">Save and New<span class="ui-icon ui-icon-disk"></span></a>')
            .click(function() {
                alert("click!");
            }).addClass("fm-button ui-state-default ui-corner-all fm-button-icon-left")
              .prependTo("#Act_Buttons>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 شده‌است.

نویسنده: Masoud.Bahrami
تاریخ: ۱۳۹۳/۰۴/۱۹ ۱۳:۲

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 افزونه ی jTable استفاده کنم. هر چند مجبور شدم فایل‌های js اون رو خیلی تغییر بدم. ولی افزونه‌ی خوبی است.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۱۹ ۱۳:۱۲

مطالب مرتبط به هم سایت جاری را از طریق برچسب‌ها و گروه‌بندی‌های ذیل مطالب بهتر می‌توانید دنبال کنید.
- ویرایش inline
- تهیه خروجی
+
لیست مجموعه مثال‌ها و قابلیت‌های تکمیلی jqGrid (اگر قصد مقایسه دارید)

نویسنده: صالح
تاریخ: ۱۳۹۳/۰۴/۲۰ ۱۸:۵۵

من از گرید شما استفاده کردم ولی متأسفانه وقتی تعداد سطرها زیاد باشه 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 ایراد چه میشه کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۲۰ ۲۰:۳۸

در مطلب فعال سازی این فرم‌ها ، متد centerDialog و نحوه‌ی اعمال آن به رخداد beforeShowForm عنوان شده. ایده‌ای هست در مورد نحوه‌ی مشخص سازی محل نمایش این فرم‌ها در صفحه.

نویسنده: Masoud Bahrami
تاریخ: ۱۳۹۳/۰۴/۲۳ ۱۴:۲۲

با تشکر
منظور امکان افزودن بود. یعنی وقتی دکمه افزودن رو میزنه یک سطر جدید واشه توگرید همونجا بنویسه مشکل اصلی من با jTable هم همینه و این یک مورد خیلی مهمه برا کاربرا که نخواد برا افزودن فرم مودال واشه براش.
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 افروزنه هم داشته باشه و معرفی کنید ممنون میشم
در هر صورت بازم ازتون تشکر می‌کنم

نویسنده: rahele
تاریخ: ۱۳۹۳/۰۴/۲۸ ۱۴:۳۲

با سلام؛ چطور می‌توانیم به تابع refresh گرید دسترسی داشته باشم تا بعد از انجام عملیات‌های مختلف , آن را فراخوانی کنم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۲۸ ۱۴:۳۸

```
$("#list").trigger("reloadGrid");
```

ولی در عمل نیازی به این‌کار 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** : false است که می‌شود آن‌ها را true 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 می باشد. محبت بفرمایید برای عملیات‌های inline راهنمایی نمایید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۳۰ ۱۱:۲۸

مراجعه کنید به مطلب « فعال سازی و پردازش Inline Add در jqGrid »

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۳۰ ۱۲:۳۵

مراجعه کنید به مطلب « فعال سازی و پردازش Inline Add در jqGrid »

عنوان: **تهیه خروجی PDF و اکسل از حاصل جستجوی پویای jqGrid به کمک PDF Report**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۱۶ ۸:۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, PdfReport, 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 کتابخانه‌ی 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 گرفته‌است؟ بر روی کدام فیلدها با چه مقادیری جستجو انجام شده‌است؟ تا ... بتوانیم بر این مبنا، منبع داده‌ی موجود را فیلتر کرده و لیست نهایی را تبدیل به گزارش کنیم. گزارشی که دقیقا با اطلاعاتی که کاربر در صفحه مشاهده می‌کند، تطابق داشته باشد.

خوشبختانه تمام این سؤالات توسط متد 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(
                // مانند قبل
                }).jqGrid('navButtonAdd', '#pager', {
                    caption: "", buttonicon: "ui-icon-print", title: "خروجی پی دی اف",
                    onClickButton: function () {
                        $("#list").jqGrid('excelExport', { url: '@Url.Action("GetProducts",
"Home")' });
                    }
                });
        });
    </script>
}
```

در اینجا توسط متد navButtonAdd یک دکمه‌ی جدید را اضافه کرده‌ایم که کلیک بر روی آن سبب فراخوانی متد excelExport و ارسال اطلاعات گزارش به url تنظیم شده‌است. باید دقت داشت که این اطلاعات از طریق Http Get به سرور ارسال می‌شوند و دقیقا 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 شده‌است. این oper در اینجا با excel مقدار دهی می‌شود.

البته چون تعداد این پارامترها بیش از اندازه شده‌است، 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 متد 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()
            };
```

```
        return Json(productsData, JsonRequestBehavior.AllowGet);
    }
```

توضیحات:

اکثر قسمت‌های این متد با متدی که 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 کتابخانه‌ی Dynamic LINQ به صورت پویا عمل می‌کند و متد ApplyFilter، کار تهیه where پویا را انجام می‌دهد.

فقط در اینجا بررسی و پردازش پارامتر oper نیز اضافه شده‌است. اگر این پارامتر مقدار دهی شده باشد، یعنی نیاز است کل اطلاعات را واکشی کرد؛ زیرا می‌خواهیم گزارش گیری کنیم و نه اینکه صرفا 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 نهایتا 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: ۱۳۹۳/۰۴/۱۷ ۹:۳۸

با سلام من هنگام اجرای پروژه با خطای زیر روبرو 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: ۱۳۹۳/۰۴/۱۷ ۹:۵۴

- ابتدا به اینترنت وصل شوید.
- سپس در خط فرمان پاورشل نیوگت دستور زیر را اجرا کنید:

```
PM> update-package -reinstall
```

به این صورت بسته‌های MVC 5 آن به صورت صحیح به پروژه اضافه می‌شوند.

نویسنده: سروش
تاریخ: ۱۳۹۳/۰۴/۱۷ ۱۷:۵۷

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: ۱۳۹۳/۰۴/۱۷ ۱۹:۰

دریافت آخرین نگارش نیوگت از اینجا ^ و ^

نویسنده: سروش
تاریخ: ۱۳۹۳/۰۴/۱۸ ۱۱:۴۶

با سلام و ادب

در پروژه خروجی Excel بوسیله EPPlus 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 اکسترکت کنم
درضمن پسوند عجیبی دارد (jqGrid06.7z)
ممنون

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

تصویر بالا تعداد هزار و پانصد می‌باشد .

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.

عنوان: **Ajax.BeginForm و ارسال فایل به سرور در ASP.NET MVC**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۲۱ ۸:۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, jQuery Ajax, File upload

Ajax.BeginForm در ASP.NET MVC از jQuery Ajax 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) فایلی را به سرور ارسال کرد. یا باید از سیلورلایت یا فلش استفاده نمود و یا از مرورگرهایی که XMLHttpRequest Level 2 را پشتیبانی می‌کنند (از IE 10 به بعد مثلا) که امکان Ajax upload 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 ایی به سرور ارسال خواهند شد. المان file آن 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 ایی به نظر می‌رسد.
برای انجام اینکار تنها کافی است از افزونه‌ی AjaxFileUpload استفاده کنیم که در ادامه نحوه‌ی استفاده از آن‌را بررسی خواهیم کرد.

پیشنیازها

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 متد 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(HttpPostedFileBase image1, int id)
        {
            var isAjax = this.Request.IsAjaxRequest();
            Thread.Sleep(3000); //شبیه سازی عملیات طولانی
            return Json(new { FileName = "/Uploads/filename.ext" }, "text/html",
JsonRequestBehavior.AllowGet);
        }
    }
}
```

Index اول، کار نمایش صفحه‌ی ارسال اطلاعات را انجام خواهد داد.

Index دوم کار پردازش Ajax ایی اطلاعات ارسالی به سرور را به عهده دارد. HttpPost آن Ajax ایی است.

متد UploadFiles، کار پردازش اطلاعات ارسالی از طرف IFrame مخفی را انجام می‌دهد. HttpPost آن معمولی 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)
    <br />
    <label>Image:</label>
    <br />
    <input type="file" name="Image1" id="Image1" />
    <br />
    <input type="submit" value="Submit" />
    <img id="loading" src="~/Content/Images/loading.gif" style="display:none;">
}

@section Scripts
{
    <script type="text/javascript">
        $(function () {
            $('#uploadForm').submit(function () {
                $("#loading").show();
                $.ajaxFileUpload({
                    url: "@Url.Action("UploadFiles", "Home")", // مسیری که باید فایل به آن ارسال شود
                    secureuri: false,
                    fileElementId: 'Image1', // آی دی المان ورودی فایل
                    dataType: 'json',
                    data: { id: 1, data: 'test' }, // اطلاعات اضافی در صورت نیاز
                    success: function (data, status) {
                        $("#loading").hide();
```

```
                        if (typeof (data.FileName) != 'undefined') {
                            alert(data.FileName);
                        }
                    },
                    error: function (data, status, e) {
                        $("#loading").hide();
                        alert(e);
                    }
                });
            });
        });
    </script>
}
```

فرمی که توسط Ajax.BeginForm تشکیل شده‌است، یک فرم معمولی Ajax ایی است و نکته‌ی جدیدی ندارد. تنها در آن یک المان ارسال فایل قرار گرفته‌است و همچنین Id آن‌را نیز جهت استفاده توسط jQuery مشخص کرده‌ایم.

در ادامه نحوه‌ی فعال سازی ajaxFileUpload را دقیقا در زمان submit فرم، مشاهده می‌کنید. در اینجا url آن به اکشن متدی که اطلاعات المان file را باید دریافت کند، اشاره می‌کند. fileElementId آن مساوی Id المان فایل فرم Ajax ایی صفحه‌است. از قسمت data جهت ارسال اطلاعات اضافه‌تری به اکشن متد UploadFiles استفاده می‌شود. سایر قسمت‌های آن نیز مشخص هستند. اگر عملیات موفقیت آمیز بود، success 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 متدهای 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

ممنون 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 میدادم اما مشکل من این بود که اول فایل آپلود میشد و بعد فرم ثبت میشد.یعنی اول رویداد submit کلاینت اجرا میشد 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 متد جاوا اسکریپتی که قرار است پس از اجرای موفقیت آمیز ارسال اطلاعات Ajax ایی به سرور 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 متد یک چنین امضایی را باید داشته باشد:

```
function doUpload(data, status, xhr) {
            alert(data.result);
            // مابقی 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 افزونه‌ی ارسال فایل، به اکشن متد ذخیره 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 متد doUpload فوق استفاده شد.

نویسنده: رضا
تاریخ: ۱۳۹۳/۰۷/۳۰ ۱۸:۷

جناب آقای نصیری ممنون از مطلبتون
در صورتیکه ارسال فایل بصورت فرم‌های مودال مثلا jquery ui و بصورت ajax بخواد 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 متد پست مربوطه به 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>
    <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.
ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۸/۳۰ ۲۱:۵۰

اصل افزونه در اینجا مطرح شده‌است. سورس آن‌را دریافت کنید و مطابق نیاز خودتان آن‌را توسعه دهید (بجای یک fileElementId که در سورس وجود دارد، آن‌را تبدیل کنید به یک آرایه مثلا).

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 Propertyهای کلاس محصول ImageName باشد، باید نام عکس ارسالی را دریافت و ذخیره کنیم، اکشن متد UpoadFiles نام عکس ذخیره شده را برگشت می‌دهد، حال به چه طریقی میتوان به این نام در اکشن متد Index دسترسی داشت؟ نام برگشتی از اکشن متد 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 متد Index، اینبار Id رکورد ثبت شده را هم داشته باشد. به این Id در متد doUpload دسترسی خواهید داشت. در ادامه کدهای

$('uploadForm').submit کلا حذف می‌شوند و متد ajaxFileUpload آن به داخل متد doUpload که در حین OnSuccess فراخوانی شده، منتقل خواهد شد. در اینجا Id را به اکشن متد 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.

نویسنده: مجید و مسعود منظوری
تاریخ: ۱۳۹۳/۱۰/۲۸ ۰:۲۹

با سلام
ما مطابق آموزشی که در این مقاله داده شده از یک اکشن متد برای ذخیره عکس ارسالی تو یک پوشه و سپس برگشت دادن مسیر عکس و از یک اکشن متد دیگه برای ذخیره اطلاعاتی که قراره همراه با فرم ارسال بشن (به همراه مسیر عکس برگشت داده شده)، استفاده میکنیم
مشکلی که ما موقع استفاده از این افزونه باهاش برخوردیم اینه که گاهی اوقات و همونطور که انتظار میره اکشن متد (AddAvatars) که وظیفه ذخیره عکس رو داره اول اجرا میشه و اکشن متد (Add) که وظیفه ذخیره اطلاعات رو داره دوم، ولی گاهی اوقات این ترتیب به هم میریزه و ابتدا اطلاعات ارسالی ذخیره میشه و بعد اکشن متد ذخیره عکس اجرا میشه.
سناریوی ما هم تا حدی شبیه به سناریویی هست که آقای احمدی مطرح کردند، ولی همونطور که گفتیم مشکل اصلی اینه که اکشن متدها هر بار با ترتیب‌های متفاوت فراخوانی میشن

```
<div class="container-fluid">
    @using (Ajax.BeginForm("Add", "Authors", new AjaxOptions { UpdateTargetId = "result", InsertionMode
= InsertionMode.Replace, HttpMethod = "POST" }, new { @class = "form-horizontal", id = "UploadFile" }))
    {
        @Html.AntiForgeryToken()

        <div class="control-group">
            <label class="control-label" for="AuhtorFirstNameAndLastName">نام نویسنده</label>
            <div class="controls">
                @Html.TextBoxFor(author => author.AuhtorFirstNameAndLastName, new { placeholder = "نام
نویسنده" })
            </div>
            @Html.ValidationMessageFor(author => author.AuhtorFirstNameAndLastName)
        </div>

        <div class="control-group">
            <label class="control-label" for="Status">ارسال عکس</label>
            <div class="controls">
                <input type="file" name="avatarFile" id="avatarFile" />
            </div>
            <div>
                @*<input type="submit" name="btn-submit" value="ارسال" class="btn btn-success" />*@
                <img id="loading" alt="1" src="Images/loading83.gif" style="display: none;" />
            </div>
        </div>

        <div id="result"></div>
        <input type="submit" name="btn-submit" value="افزودن نویسنده" class="btn btn-success" />
        <input type="button" name="btn-colose" id="btn-close" value="انصراف" class="btn btn-danger"
onclick="$dialog.dialog('close');" />
    }
</div>

<script type="text/javascript">
        $('#UploadFile').submit(function () {
            $("#loading").show();
            $.ajaxFileUpload({
                url: "@Url.Action("AddAvatar","Authors")",
                 secureuri: false,
                 fileElementId: 'avatarFile',
                 dataType: 'json',
                 data: {},
                 success: function (data, status) {
                     $("#loading").hide();
                 },
                 error: function (data, status, e) {
                     $("#loading").hide();
                 }
             });
```

```
        });
</script>
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۲۸ ۰:۳۶

کمی بالاتر پاسخ دادم: «...در ادامه کدهای $('uploadForm').submit *کلا حذف می‌شوند* و متد ajaxFileUpload آن به *داخل* متد doUpload که در حین OnSuccess فراخوانی شده، منتقل خواهد شد. .. »

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۲۸ ۱:۱۱

جهت اطلاع

مثال تکمیلی این بحث به همراه نکات مرتبط با OnSuccess ، بازگشت Id رکورد پس از ثبت و ارسال آن به اکشن متد آپلود، به مخزن کد ذیل منتقل شدند:

MVC-Ajax-Form-Upload

عنوان: آپلود فایل توسط فرم‌های پویای jqGrid
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۲۴ ۱۱:۳۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, jqGrid

پیشنیازها

Ajax.BeginForm و ارسال فایل به سرور در ASP.NET MVC
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 ازExpressionها جهت ایجاد Strongly typed view در ASP.NET MVC

فرم‌های پویای jqGrid نیز به صورت Ajax ایی به سرور ارسال می‌شوند و اگر نوع عناصر تشکیل دهنده‌ی آن‌ها file تعیین شوند، قادر به ارسال این فایل‌ها به سرور نخواهند بود. در ادامه نحوه‌ی یکپارچه سازی افزونه‌ی AjaxFileUpload را با فرم‌های jqGrid بررسی خواهیم کرد.

تغییرات فایل Layout برنامه

در اینجا دو فایل جدید ajaxfileupload.js و jquery.blockUI.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 واکنشگرا 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="rsperror"></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 متد توکار 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 متد 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/', '');
                        }
                    }
                ],
```

edittype 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 مقدر 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.
Rnd اضافه شده به انتهای آدرس تصویر، جهت جلوگیری از کش شدن آن تعریف شده‌است.

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 StronglyTyped با متد PropertyName جهت دریافت نام رشته‌ای یک خاصیت تعریف شده‌است. گاهی از اوقات این تنها چیزی است که کدهای سمت کلاینت، جهت سازگار شدن با Refactoring و Strongly typed تعریف شدن نیاز دارند و نه ... محصور کننده‌هایی 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 کننده‌هارا پیدا کنید اما ... هیچکدام کامل نیستند و دست آخر مجبور خواهید شد در بسیاری از موارد مستقیما JavaScript نویسی کنید.

یکپارچه سازی افزونه‌ی AjaxFileUpload با فرم‌های jqGrid

پس از این مقدمات، ستون ویژه‌ی actions که inline edit را فعال می‌کند، چنین تعریفی 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 شده‌است تا کار ارسال فایل به سرور را در حالت ویرایش inline فعال کند.

و ویژگی‌های قسمت‌های add، edit و delete 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 MVC » آشنا شده‌اید. تنها قسمت جدید آن شامل رویدادگردان afterSubmit است. در اینجا است که افزونه‌ی AjaxFileUpload

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 کرده‌ایم.

در ادامه تعاریف متدهای doInlineUpload و doUpload بکار 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, "لطفا فایلی را انتخاب کنید", result.id];
            }

            $('#grid1').block({ message: '<h4>در حال ارسال فایل به سرور</h4>' });
            $.ajaxFileUpload({
                url: "@Url.Action("UploadFiles", "Home")", // مسیری که باید فایل به آن ارسال شود
                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 رویدادگردان‌های afterSubmit و afterSave یکی نیست. به همین جهت دو متد اضافی به جای یک متد doUpload مورد استفاده قرار گرفته‌اند.

متد doUpload توسط 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 شده‌است یا خیر؟ اگر خیر، یک پیام اعتبارسنجی سفارشی به او نمایش داده خواهد شد.

خروجی متد doUpload حتما 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 فراخوانی متداول افزونه‌ی ajaxFileUpload را مشاهده می‌کنید. تنها نکته‌ی مهم آن فراخوانی متد reloadGrid در حالت success است. به این ترتیب گرید را وادار می‌کنیم تا اطلاعات ذخیره شده در سمت سرور را دریافت کرده و سپس تصویر را به نحو صحیحی نمایش دهد.

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 دقت کنید، این id را دریافت می‌کند. بنابراین می‌توان یک ربط منطقی را بین فایل ارسالی و رکورد متناظر با آن برقرار کرد.
Content type مقدار بازگشتی از متد UploadFiles حتما 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: ۱۳۹۳/۰۵/۰۴ ۲۲:۱۱

سلام
strongly typeها فقط نام پروپرتی‌ها را بر می‌گرداند اگر بخواهیم مقدار یکی از data annotationها مثل display برگرداند و یا اگر بخواهیم از validationهای data annotationها استفاده کنیم باید چه کار کنیم .ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۰۵ ۱:۱۰

- به پیشنیازهای بحث مراجعه کنید؛ مورد آخر .
- مستقل هست از مباحث اعتبارسنجی سمت کاربر تعریف شده توسط data annotationها. از افزونه‌ی jQuery Validator استفاده نمی‌کند. سیستم خاص خودش را دارد. فقط از اعتبارسنجی سمت سرور حاصل data annotationها می‌شود در آن استفاده کرد .

عنوان: **اختصاصی کردن Razor برای #C در MVC با استفاده از Extension Method**
نویسنده: محمد محمدصادقی
تاریخ: ۱۳۹۳/۰۴/۲۵ ۱۳:۱۰
آدرس: 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 Viewهایی 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.

Extension Methodها بسیار کارآمد هستند و نحوه ایجاد و استفاده از آنها بسیار راحت است. به گونه ای که میتوان آنها را حتی برای کلاس‌های 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 متدی 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 متد مورد نظر را مینویسیم:

```
namespace ApplicationTest01.Utilities
{
    public class StringHelper
    {
        public string StringConcatenate(string firstPhrase, string secondPhrase)
        {
            return firstPhrase + secondPhrase;
        }
    }
}
```

3. کلاس و متدی که در آن تعریف میکنیم بایستی public و static باشند. namespace کلاس را نیز به namespace کلاس string در .Net (یعنی System) تغییر میدهیم.

```
namespace System
{
    public static class StringHelper
    {
        public static string StringConcatenate(string firstPhrase, string secondPhrase)
        {
            return firstPhrase + secondPhrase;
        }
    }
}
```

4. در Extension Methodها ورودی اول تابع به پارامتری اشاره دارد که قرار است هنگام استفاده از آن (یا صدا زدن آن) متد، عملیات 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 namespace کلاس RazorViewEngine (یعنی System.Web.Mvc) یکی باشد. خروجی Extension Method ما از نوع RazorViewEngine 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 viewها شامل Peroperty هایی از جنس []string می‌باشد، ابتدا متدی ساده به نام FilterOutVbhtml برای فیلتر کردن stringهای 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 پروپرتی‌های RazorViewEngine را فرا خوانی 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 یکبار ViewEngineها را حذف میکنیم -Engine مربوط به aspx به صورت پیش فرض فعال میباشد- و سپس RazorViewEngine را به همراه فراخوانی Extension Method خودمان به ViewEngineها اضافه 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 ۱:۳۰

دوست عزیز با تشکر از مقاله شما.اما می‌خواستم بدون مورد استفادش چیه و کجا به کارمون میاد؟

نویسنده: محمد محمدصادقی
تاریخ: ۱۳۹۳/۰۴/۲۶ ۱۱:۸

در حالت عادی RazorViewEngine هنگام Load کردن یک View 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، Engine 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 میتونه موثر باشه

سوال دیگه هست در خدمتم

نویسنده: سید مهران موسوی
تاریخ: ۱۳۹۳/۰۴/۲۶ ۱۹:۵

دوست عزیز برای جلوگیری از Lookuping سایر Viewهای مرتبط با ViewEngineهای دیگه (پسوند aspx) نیاز به این همه کار اضافه نیست . کافیه سایر ViewEngine هارو حذف کنید.

```
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

RazorViewEngine 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: ۱۳۹۳/۰۴/۲۷ ۱۹:۴۵

طبق مطالعات بنده 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 پسوند vbhtml رو 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: ۱۳۹۳/۰۴/۲۸ ۱۶:۴

من کدهای شمارو اجرا کردم و خروجی زیر رو گرفتم

در تصویر بالا razor جلوی Lookuping پسوندهای vbhtml رو نگرفته. لطفا شما هم امتحان کنید ببینید خروجی دیگه ای میگیرید یا همین خروجی برای شما هم میاد؟

عنوان: **روشی سریع برای ایجاد RSS و Sitemap در ASP.NET MVC**
نویسنده: مرتضی دلیل
تاریخ: ۱۳۹۳/۰۴/۲۵ ۲۱:۲۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, RSS, sitemap

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 ببندید.

پیش از تشریح مراحل، به ساختار این دو فایل توجه کنید.

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 متدهای 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()
        {
          // یک کوئری که لیستی از تایپ مشخص شده به ما بدهد
        }

         public IEnumerable<PostToXml> GetLinks()
        {
            // یک کوئری که لیستی از تایپ مشخص شده به ما بدهد
        }
```

این کنترلر دارای دو اکشن متد Rss و Sitemap است و این اکشن‌ها وظیفه‌ی ایجاد فایل‌های Xml را به عهده دارند. مواد اولیه این xml ها از دو متد GetRssFeed و GetLinks تهیه می‌شوند. 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 می‌خواهیم در Rss یا Sitemap ثبت شوند، تهیه کنیم. این لیست بر اساس شکل تنظیم دیتابیس و مسیریابی سایت، می‌تواند پیچیده و یا ساده باشد. (به کمک کوئری گرفتن با linq و یا اضافه کردن مستقیم آدرس‌ها 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 کرده‌اید دسترسی داشته باشید :

```
http://domain.com/rss
http://domain.com/sitemap
```

فایل پروژه را دریافت کنید :

MVC_RSS_Sitemap-43ad3c6681734b34b91deaaabcdba871.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: میثم 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 خواهید داشت. مهم هم نیست که اطلاعات این لیست از جدولی تهیه می‌شود یا صرفا با متد Add اضافه شده‌اند (استفاده کننده از آن کاری به منبع داده ندارد).
نهایتا برای یکی کردن چندین لیست PostToXML به یک لیست PostToXML جهت استفاده در این متدها، از متد Concat و یا Union استفاده کنید:

```
List<PostToXML> total = list1.Concat(list2);
```

نویسنده: میثم سام
تاریخ: ۱۳۹۳/۱۱/۲۱ ۲۲:۴۶

سلام؛ من یه سایت دارم مینویسم که همش دینامیک و یه سری شرکت توش هست. میخوام مثلا اسم این شرکت‌ها و لینک هاشونو از توی دیتابیس بخونم و توی سایت مپ قرار بدم. لازم به ذکر که این شرکت‌ها هر کدومشون توی همین سایت یه صفحه برای خودشون دارن. من کل مبحث سایت مپ متوجه اش شدم ولی یه چیز اصلا نفهمیدم. کجا باید این تابع فرخوانی کرد. مثلا وقتی این شرکت داره اضافه میشه داخل دیتابیس . یا وقتی که میخوایم این صفحه خود شرکت باز بشه. یا نه وقتی خود لینک مثلا : http://www.domain.com/sitemap
میزنیم همون موقع باید از دیتابیس اسم شرکت‌ها رو بخونیم و لینکشونو وارد سایت مپ کنیم و هیچ کاری به کنترل‌های دیگه نداره

و این که توی سایت خوبتون چند تا روش برای سایت مپ گفته شده به نظر خودتون کدوم یک بهتره
خیلی ممنونم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۱/۲۱ ۲۲:۵۷

- «کجا باید این تابع فرخوانی کرد»
دقیقا مطابق مثالی که پیوست کردند، در اکشن متدی به همین نام. کار بیشتری هم لازم نیست انجام شود. هر زمانیکه یک موتور جستجو به سایت شما رسید، این آدرس را واکشی می‌کند (اینکه از کجا باید بداند این آدرس را نیاز است واکشی کند، مرتبط است به مباحث فایل robots.txt و قید صریح آدرس آن). پس از واکشی خودکار آن، از بانک اطلاعاتی گزارش تهیه کرده و لیستی از PostToXML را مطابق این مطلب بازگشت می‌دهید.
- « به نظر خودتون کدوم یک بهتره »
فرقی نمی‌کنند. خروجی نهایی یک استاندارد بیشتر ندارد.

عنوان: **اعتبارسنجی سفارشی سمت کاربر و سمت سرور در jqGrid**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۲۷ ۱۲:۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, jqGrid

همانطور که در مطلب « فعال سازی و پردازش صفحات پویای افزودن، ویرایش و حذف رکوردهای jqGrid در ASP.NET MVC » نیز ذکر شد، خاصیت editrules یک ستون، برای مباحث اعتبارسنجی اطلاعات ورودی توسط کاربر پیش بینی شده‌است. 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، minValue و maxValue از نوع عددی، email, url, date, time از نوع bool و custom قابل تنظیم است. در ادامه نحوه‌ی 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 3 تا 40 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.

اعتبار سنجی 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 3 تا 40 حرف باشد"];
                                else
```

```
                                return [true, ""];
                            }
                        }
                    },
                ],
```

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 آن‌را بررسی نمود. این‌کار، توسط مقدار دهی خاصیت custom به true و سپس تعریف متدی برای custom_func قابل انجام است.

خروجی این متد یک آرایه دو عضوی است. اگر عضو اول آن true باشد، یعنی اعتبارسنجی موفقیت آمیز بوده‌است؛ اگر خیر، عضو دوم آرایه، پیامی است که به کاربر نمایش داده خواهد شد.

تا اینجا کار اعتبارسنجی سمت کاربر به پایان می‌رسد. اما نیاز است در سمت سرور نیز بررسی شود که آیا نام وارد شده تکراری است یا خیر. برای این منظور تنها کافی است رویداد afterSubmit حالت‌های Add و Edit را بررسی کنیم:

```
$('#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) {
                //نمایش خطای 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 ایی را در حالت‌های شکست اعتبارسنجی 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);
            }
```

```
            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.

اعتبار سنجی 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 required: true، کار تنظیم ورود اجباری ایمیل کاربر به پایان می‌رسد. همچنین با تنظیم email: 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.

اعتبار سنجی 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 این حالت خاص قابل استفاده نیست
                            //در حالت ویرایش رکورد، ورود کلمه عبور اختیاری است
                            //در حالت افزودن رکورد، ورود کلمه عبور اجباری است
                        }
                    },
                ],
```

حالت بررسی اعتبارسنجی کلمه‌ی 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 روال رویدادگردان 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, "لطفا کلمه عبور را وارد کنید"];
                        return [true, ""];
                    }
                },
                //delete options
                {
                });
        });
```

چون می‌خواهیم تنها حالت Add را تحت کنترل قرار دهیم، رویدادگردان beforeSubmit آن‌را مقدار دهی کرده‌ایم. توسط postdata کلیه اطلاعات قابل ارسال به سرور به صورت یک شیء جاوا اسکریپتی یا JSON در اختیار ما است. سپس با بررسی برای مثال postdata.Password می‌توان در مورد مقدار کلمه‌ی عبور تصمیم گیری کرد. در اینجا نیز خروجی متد باید یک آرایه دو عضوی باشد تا در صورت false بودن اولین عضو آن، پیام سفارشی اعتبارسنجی خاصی را بتوان به کاربر نمایش داد.

اعتبار سنجی 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 "<a href='" + cellvalue + "' >" + cellvalue + "</a>";
                        },
                        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 فرمت URL وارد شده نیز بررسی می‌شود.

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید

jqGrid08.zip

نظرات خوانندگان

نویسنده: داوود
تاریخ: ۱۳۹۳/۰۴/۲۷ ۱۵:۳۲

باسلام
بعداز دانلود فابل 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
پیشاپیش 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 گزینه‌ی Advanced save options آن‌را 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 می‌تونیم عمل کنیم منظورم به یک کاربر اجازه ثبت بدیم و یا فقط بتونه اطلاعات را ببینه 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>
colMdl.push(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 درقالب یک مثال کامل توضیح بدهید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۰۱ ۱۰:۲۲

جهت مباحث تکمیلی اعتبارسنجی کاربران به این مطالب مراجعه کنید:

- اعتبار سنجی کاربران در ASP.NET MVC
- مدیریت سفارشی سطوح دسترسی کاربران در MVC
- Iris Membership برای احراز هویت کاربران در ASP.NET MVC به صورت پویا
- ASP.NET Identity

عنوان: OutputCache در ASP.NET MVC
نویسنده: مرتضی دلیل
تاریخ: ۱۳۹۳/۰۴/۲۸ ۱۴:۱۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, Performance, Caching

مقدمه

OutputCaching باعث می‌شود خروجیِ یک اکشن متد در حافظه نگهداری شود. با اعمال این نوع کشینگ، ASP.NET در خواست‌های بعدی به این اکشن را تنها با بازگرداندن همان مقدار قبلیِ نگهداری شده در کش، پاسخ می‌دهد. در حقیقت با OutputCaching 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 کتگوری‌های موجود در سایت.

روش‌های مختلفی برای کوئری گرفتن وجود دارد، به عنوان مثال ما به کمک یک یا چند کوئری و توسط یک ViewModel جامع، میخواهیم اطلاعات را به سمت ویو ارسال کنیم. پس در اکشن متد Index ، حجم تقریبا کمی از اطلاعات را باید به کمک کوئری(کوئری های) تقریبا 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 کوئری (کوئری های) مورد نظر زیاد صدا زده میشوند ، در حقیقت صفحه اول احتمالا بیشترین فشار ترافیکی را در بین صفحات ما دارد، البته این فقط یک احتمال است و ما دقیقا از این موضوع اطلاع نداریم.

یکی از راه‌های انجام یک کش موفق و دانستن لزوم کش کردن، این است که دقیقا بدانیم ترافیک سایت روی چه صفحه ای بیشتر است. در واقع باید بدانیم در کدام صفحه "هزینه‌ی اجرای عملیات موجود در کد" بیشترین است.

فشار ترافیکی(ریکوئست‌های زیاد) و آپدیت‌های روزانه‌ی دیتابیس را، در دو کفه ترازو 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 گاهِ شما، در پاسخِ ریکوئست‌ها دیده شوند. کمی کفه‌ی کش کردن را سنگین 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 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 متد، 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. بار دیگر صفحه را رفرش کنیم، **اگر این "بار دیگر" در کمتر از 60 ثانیه پس از رفرش قبلی اتفاق افتاده باشد برنامه روی 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 ریکوئست هایی 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 عملیاتِ هزینه دار(مثل کوئری‌های حجیم) 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 تغییر 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 نگذرد، سایت آپدیت نخواهد شد و محتوا همان چیزهای قبلی باقی خواهند ماند. اما چاره چیست؟

کافیست در بخش ادمین، وقتی که یک پست ایجاد میکنید یا پستی را ویرایش میکند 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 OutputCatch دارای پارامترهای زیادیست و در این مقاله فقط به توضیح عملکرد این اتریبیوت اکتفا شده است. بطور کلی این مبحث ظاهر ساده ای دارد، ولی نحوه استفاده از کشینگ کاملا 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 کمک 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 نمیدانید) پس مجبور به آزمون و خطا و تست و اندازه گیری هستید تا بدانید چه مقدار زمانی را برای چه پروفایلی قرار دهید. مثلا 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.

[اینجا] و [اینجا]

نظرات خوانندگان

نویسنده: ابوالفضل رجب پور
تاریخ: ۱۳۹۳/۰۴/۲۹ ۹:۵۳

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: ۱۳۹۳/۰۴/۲۹ ۱۰:۲۴

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: ۱۳۹۳/۰۴/۲۹ ۱۰:۲۶

- با استفاده از jQuery که یک بحث سمت کاربر است، زمانیکه صفحه نمایش داده شد، یک درخواست Ajax ایی به اکشن متدی 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: ۱۳۹۳/۰۸/۱۲ ۱۶:۵۲

با سلام.
متدی 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 متد 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: ۱۳۹۳/۰۸/۱۲ ۱۸:۴۱

آیا از Html.RenderAction برای نمایش آن استفاده کردید؟ اگر بله، متد یاد شده تاثیری روی کش آن نداره، چون نحوه‌ی کش شدن child actionها متفاوته.

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۳/۰۸/۱۲ ۱۹:۱۵

بله. راه حل مشکل چیست؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۸/۱۳ ۱۵:۱۴

به این صورت؛ البته این روش کش تمام child actionها را با هم پاک می‌کند:

```
OutputCacheAttribute.ChildActionCache = new MemoryCache("NewRandomStringNameToClearTheCache");
```

عنوان: فعال سازی و پردازش Inline Add در jqGrid
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۳۰ ۸:۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, jqGrid

پیشنیازها

فعال سازی و پردازش صفحات پویای افزودن، ویرایش و حذف رکوردهای jqGrid در ASP.NET MVC
اعتبارسنجی سفارشی سمت کاربر و سمت سرور در jqGrid

پیشتر با نحوه‌ی فعال سازی صفحات پویای افزودن، ویرایش و حذف رکوردهای jqGrid آشنا شدیم. اما ... شاید علاقمند نباشید که اصلا از این صفحات استفاده کنید. شاید به نظر شما با کلیک بر روی دکمه‌ی + افزودن یک رکورد جدید، بهتر باشد داخل خود گرید، یک سطر خالی جدید باز شده تا بتوان آن‌را پر کرد. شاید این نحو کار کردن با گرید، 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 کرده بودیم. 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 save و cancel 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',
//...
)
                .jqGrid('gridResize', { minWidth: 400, minHeight: 150 })
                .jqGrid('inlineNav', '#pager',
                {
                    edit: true, add: true, save: true, cancel: true,
                    edittext: "ویرایش", addtext: "جدید", savetext: "ذخیره", canceltext: "لغو",
                    addParams: {
                        // اگر می‌خواهید ردیف‌های جدید در ابتدا ظاهر شوند، این سطر را حذف کنید
                        position: "last", //ردیف‌های جدید در آخر ظاهر می‌شوند
                        rowID: '_empty',
                        useDefValues: true,
```

```
                    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 گرید اضافه شده‌است.

در اینجا 4 دکمه‌ی ویرایش، جدید، ذخیره و لغو، در نوار pager پایین گرید ظاهر خواهند شد (سمت چپ؛ سمت راست همان دکمه‌های نمایش فرم‌های پویا هستند).

سپس باید دو قسمت مهم addParams و editParams آن‌را 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 شده‌است.

position در اینجا به معنای محل افزوده شدن یک ردیف خالی است. مقدار پیش فرض آن first است؛ یعنی همیشه در اولین ردیف گرید، این ردیف جدید اضافه می‌شود. در اینجا به last تنظیم شده‌است تا در پایین گرید و پس از رکوردهای موجود، نمایش داده شود.

useDefValues سبب استفاده از مقادیر پیش فرض تعریف شده در ستون‌های گرید در حین افزوده شدن یک ردیف جدید می‌گردد.

addRowParams و editParams هر دو ساختار تقریبا یکسانی دارند که به نحو ذیل تعریف می‌شوند:

```
function getInlineNavParams(isAdd) {
            return {
                // استفاده از آدرس‌های مختلف برای حالات ویرایش و ثبت اطلاعات جدید
                url: isAdd ? '@Url.Action("AddUser", "Home")' : '@Url.Action("EditUser","Home")',
                key: true,
                restoreAfterError: false, // این مورد سبب می‌شود تا اعتبارسنجی سمت سرور قابل اعمال
شود
                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 بسیار مهم است. اگر در سمت سرور خطای اعتبارسنجی گزارش شود و restoreAfterError مساوی true باشد (مقدار پیش فرض)، کاربر مجبور خواهد شد اطلاعات را دوباره وارد کند.
در روال رویدادگران oneditfunc دکمه‌ی save و cancel ردیف را که مخفی هستند، ظاهر می‌کنیم (مکمل formatter ستون action است).
در قسمت successfunc، پس از پایان موفقیت آمیز کار، متد 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 متد 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; //این مورد برای افزودن داخل ردیف‌های گرید لازم است
                return Json(new { success = false, message = "نام کاربر تکراری است" },
JsonRequestBehavior.AllowGet);
            }
```

روال رویداد گردان errorfunc، اگر مقدار 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 دهی this.Response.StatusCode را 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 خواهند کرد. برای رفع این مشکل فقط کافی است روال 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 می‌شود
        var result = $.parseJSON(response.responseText);
        if (result.success === false) {
            return result.message;
        }
        return "لطفا ورودی‌های وارد شده را بررسی کنید";
    }
```

errorTextFormat تنها در حالتیکه StatusCode بازگشتی از طرف سرور مساوی 200 نیست، فراخوانی می‌شود. در اینجا می‌توان response دریافتی را آنالیز و سپس پیام خطای سفارشی آن‌را جهت نمایش در فرم‌های پویای گرید، بازگشت داد.

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید:

jqGrid09.zip

عنوان: **بهینه سازی سرعت یافت ویوها با سفارشی سازی Lookup Caching در Razor View Engine**
نویسنده: سید مهران موسوی
تاریخ: ۱۳۹۳/۰۵/۰۲ ۹:۴۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, Razor, Performance, Caching

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 عملا بازده سیستم را تا حدودی بهبود ببخشیم.

طی مطالعاتی که بنده بر روی سورس MVC داشتم، به صورت پیش فرض، در زمانیکه پروژه در حالت Release اجرا می‌شود، نتیجه حاصل از یافت آدرس فیزیکی ویوهای متناظر با اکشن متدها 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.

نکته ای که وجود دارد این هست که علاوه بر مفید بودن این امر و بهبود سرعت در درخواست‌های متوالی برای اکشن متدها، این عمل با توجه به مشاهدات بنده 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 انقضاء مداخل موجود در حافظه‌ی کش اختصاص داده شده به Lookup Cahching و مدیریت مسائل thread-safe و ... را شامل می‌شود.

همانطور که می‌دانید، معمولا تعداد ویوها اینقدر زیاد نیست که Caching نتایج یافت مسیر فیزیکی view ها، حجم زیادی از حافظه Ram را اشغال کند پس با این وجود به نظر می‌رسد که اشغال کردن این میزان اندک از حافظه در مقابل بهبود سرعت، قابل چشم پوشی است و سیاست‌های توکار نامبرده فقط عملا تاثیر منفی در روند Lookup Caching پیشفرض MVC خواهند گذاشت. برای جلوگیری از تاثیرات منفی سیاست‌های نامبرده و عملا بهبود سرعت Caching نتایج Lookup آدرس فیزیکی ویوها میتوانیم یک لایه Caching سطح بالاتر به View Engine اضافه کنیم .

خوشبختانه تمامی View Engineهای MVC شامل Web Forms و Razor از کلاس VirtualPathProviderViewEngine مشتق شده‌اند که نکته مثبت که توسعه Caching اختصاصی نامبرده را برای ما مقدور می‌کند. در اینجا خاصیت (Property) قابل تنظیم ViewLocationCache از نوع IViewLocationCache هست .

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، حتما و حتما 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: ۱۳۹۳/۰۵/۰۲ ۹:۵۶

ضمن تشکر از ایده‌ای که مطرح کردید. طول عمر httpContext.Items فقط محدوده به یک درخواست و پس از پایان درخواست از بین می‌ره. مثلا یکی از کاربردهاش ذخیره اطلاعات Unit of work در طول یک درخواست هست و بعد از بین رفتن خودکار آن. بنابراین در این مثال cache.GetViewLocation اصلی بعد از یک درخواست مجددا فراخوانی میشه، چون GetRequestCache نه فقط طول عمر کوتاهی داره، بلکه اساسا کاری به key متد GetViewLocation نداره. کار s_key تعریف شده عموما تعریف lock هست نه استفاده ازش به عنوان کلید دیکشنری. بنابراین اگر خود MVC از HttpContext.Cache استفاده کرده، کار درستی بوده، چون به ازای هر درخواست نیازی نیست مجددا محاسبه بشه.

نویسنده: سید مهران موسوی
تاریخ: ۱۳۹۳/۰۵/۰۲ ۱۲:۲۱

ممنون از توجهتون ، بله من اشتباها httpContext.Items رو به کار برده بودم . کد موجود در مقاله اصلاح شد

نویسنده: حامد سبزیان
تاریخ: ۱۳۹۳/۰۵/۰۲ ۱۸:۴

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 Viewها حساس نیست.

عنوان: **گروه بندی اطلاعات در jqGrid**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۰۴ ۰:۲۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, jqGrid

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 مشاهدات هر گروه قابل محاسبه شود.
- جمع کل هر گروه در زمانیکه هر گروه نیز بسته‌است نمایش داده شود.
- رنگ ردیف جمع کل قابل تنظیم باشد.

آزمایش دهم

شماره	عنوان	تعداد بار مشاهده
گروه 1 - 5 ردیف		
1	عنوان 1	230
17	عنوان 17	180
37	عنوان 37	926
42	عنوان 42	322
48	عنوان 48	849
	خلاصه	جمع مشاهدات: 2507
گروه 2 - 5 ردیف		
	خلاصه	جمع مشاهدات: 2951
گروه 3 - 5 ردیف		
	خلاصه	جمع مشاهدات: 2461
گروه 4 - 5 ردیف		
	خلاصه	جمع مشاهدات: 3498

نمایش 1 - 20 از 50 صفحه 1 از 3 20

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 : ['<b>{0} - {1} ردیف</b>'],
                    groupDataSorted: true,
                    groupColumnShow: false,
                    groupCollapse: true,
                    groupSummary: [true],
                    showSummaryOnHide: true
                }
            });
```

grouping: true سبب می‌شود تا گروه بندی فعال شود. سپس نیاز است ستون یا ستون‌هایی را که قرار است بر روی آن‌ها گروه بندی صورت گیرد، معرفی کنیم. اینکار توسط خواص شیء groupingView انجام می‌شوند.

groupField بیانگر آرایه‌ای از ستون‌هایی است که قرار است بر روی آن‌ها گروه بندی صورت گیرد.

groupOrder آرایه‌ای اختیاری از مقادیر asc یا desc است که متناظر هستند با نحوه‌ی مرتب سازی پیش فرض ستون‌های معرفی شده در آرایه groupField.

groupText آرایه‌ای اختیاری از عناوین گروه بندی‌های انجام شده‌است. اگر ذکر شود، {0} آن با نام گروه و {1} آن با تعداد عناصر گروه جایگزین می‌شود.

تنظیم groupDataSorted سبب خواهد شد تا نام ستونی که بر روی آن گروه بندی صورت می‌گیرد، به سرور ارسال شود (توسط پارامتر sidx). به این ترتیب در سمت سرور می‌توان اطلاعات را به صورت پویا مرتب سازی کرده و بازگشت داد.

با تنظیم groupColumnShow به false سبب خواهیم شد تا ستون‌های معرفی شده در قسمت groupField نمایش داده نشوند.

با تنظیم groupCollapse به true، در ابتدای نمایش گروه‌ها، ردیف‌های آن‌ها نمایش داده نخواهند شد و در حالت جمع شده قرار می‌گیرند.

groupSummary به معنای فعال سازی نمایش ردیف محاسبه‌ی summary مانند sum، min، max و امثال آن بر روی یک گروه است.

اگر مقدار showSummaryOnHide مساوی true باشد، ردیف محاسبه‌ی summary حتی در حالت groupCollapse: true نمایش داده خواهد شد.

فعال سازی محاسبه‌ی جمع ستون تعداد بار مشاهدات مطالب

برای فعال سازی نهایی محاسبه‌ی 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: '<b>0} :جمع مشاهدات}</b>'
                    }
                ],
```

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.

summaryType 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 groupingRemove 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

عنوان: نمایش Subgrid در jqGrid
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۰۷ ۱۱:۲۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, jqGrid

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.

فعال سازی Subgrid

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: [{
                    name: ['شرکت', 'آدرس', 'کد پستی', 'شهر', 'کشور', 'تلفن', 'وب سایت'],
                    width: [100, 100, 100, 100, 100, 100, 100],
                    align: ['center', 'center', 'center', 'center', 'center', 'center', 'center'],
                    params: ['@(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.

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، صرفا 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 شکل فوق).

subGridOptions یک سری تنظیمات اضافه را به همراه دارد. اگر می‌خواهید اطلاعات زیرگرید فقط یکبار بارگذاری شود و با هربار کلیک کاربر از سرور دریافت نگردد، خاصیت reloadOnExpand آن‌را false کنید.

subGridUrlآدرسی 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: ۱۳۹۳/۰۵/۰۷ ۱۴:۲۱

سلام
ممنون 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.ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۰۸ ۱۲:۵۵

« ایجاد زیر گریدهای چند سطحی در jqGrid »

عنوان: **ایجاد زیر گریدهای چند سطحی در jqGrid**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۰۸ ۱۲:۵۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, jqGrid

همانطور که در مطلب ایجاد زیرگریدها در jqGrid مشاهده کردید، هرچند 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 زیرگرید را با یک گرید کامل jqGrid نیز جایگزین کرد. همچنین اگر نیاز بود، این گرید جدید چون یک jqGrid کامل است، باز هم می‌توان یک سطح دیگر را به آن افزود و الی آخر.

http://localhost:3512/H Index - My ASP.NET A...

File Edit View Favorites Tools Help

آزمایش دوازدهم

		شماره	تاریخ	تعداد اقلام	جمع کل
1	−	1	1393/05/08	30	$43,684.00

ریز اقلام سفارش 1

	شماره	محصول	واحد	قیمت
1	1	محصول 1394	7	$1,192.00
2	2	محصول 165	2	$1,049.00
3	3	محصول 867	9	$1,218.00
4	4	محصول 291	9	$1,682.00
5	5	محصول 302	3	$1,487.00
6	6	محصول 620	8	$1,665.00
7	7	محصول 1014	3	$1,444.00
8	8	محصول 1159	8	$1,400.00
9	9	محصول 804	2	$1,127.00
10	10	محصول 1305	8	$1,551.00

نمایش 1 - 10 از 30 صفحه 1 از 3 10

2	+	2	1393/05/08	30	$43,684.00
3	+	3	1393/05/08	30	$43,684.00
4	+	4	1393/05/08	30	$43,684.00
5	+	5	1393/05/08	30	$43,684.00
6	+	6	1393/05/08	30	$43,684.00
7	+	7	1393/05/08	30	$43,684.00
8	+	8	1393/05/08	30	$43,684.00
9	+	9	1393/05/08	30	$43,684.00
10	+	10	1393/05/08	30	$43,684.00

نمایش 1 - 10 از 100 صفحه 1 از 10 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",
                //..........مانند قبل
                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" })'
                      .replace("js-id", encodeURIComponent(rowId)); // تزریق اطلاعات سمت کاربر به
خروجی سمت سرور

            $("#" + subgridTableId).jqGrid({
                caption: "ریز اقلام سفارش " + rowId,
                autoencode: true, //security - anti-XSS
                url: url,
                //..........مانند قبل
            });
        }
```

همانند نمایش subgridهای معمولی، ابتدا subGrid: true باید اضافه شود تا ستونی با ردیف‌های + دار، ظاهر شود. اینبار توسط روال رویدادگردان subGridRowExpanded، کنترل نمایش subgrid را در دست گرفته و آن‌را با یک jqGrid جایگزین می‌کنیم.
امضای متد gridlRowExpanded، 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 jgGrid 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 متدی که قرار است اطلاعات زیر گرید 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 Url.Action یک کد سمت سرور است و rowId یک متغیر سمت کاربر. نمی‌توان این متغیر را در کدهای Razor مستقیما قرار داد. اما می‌توان یک محل جایگزینی را در کدهای سمت سرور پیش بینی کرد. مثلا js-id. زمانیکه این رشته در صفحه رندر می‌شود، به صورت معمول و به کمک متد replace جاوا اسکریپت، js-id 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

عنوان: **نمایش ساختارهای درختی توسط jqGrid**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۰۹ ۱۵:۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, jqGrid

jqGrid از نمایش دو ساختار درختی Nested Set model و Adjacency 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; }
    }
}
```

ParentId که سبب تولید یک مدل خود ارجاع دهنده می‌شود.
IsNotExpandable 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 jqGrid، در حالت نمایش درختی، مباحث صفحه بندی و مرتب سازی غیرفعال هستند و کدهای مرتبط با آن که در اینجا ذکر شده‌اند، فعلا 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 جی‌کیوگرید صفحه بندی کار نمی‌کند و فعال نیست و //
محاسبات 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.IsNotExpandable,
           comment.IsExpanded
                                }
                })).ToList()
            };
            return Json(productsData, JsonRequestBehavior.AllowGet);
        }
    }
}
```

nodeid اگر نال بود، یعنی کل اطلاعات ریشه‌ها (مواردی که parentId مساوی نال دارند)، باید واکشی شوند. اگر nodeid مقدار داشت، یعنی فرزند نود جاری قرار است بازگشت داده شود.

n_level مقدار جلو رفتگی نمایش اطلاعات یک نود را مشخص می‌کند. در اینجا چون با کلیک بر روی هر نود، فرزند آن از سرور واکشی می‌شود و lazy 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 4 خاصیت ساختار درختی باید به ترتیب بازگشت داده شوند.

آزمایش سیزدهم			
	شماره	نظر	تاریخ
1	1	نظر ریشه 1	1393/05/09
1	2	پاسخ 1 به ریشه 1	1393/05/09
1	3	پاسخ 1 به پاسخ 2	1393/05/09
2	4	پاسخ 2 به ریشه 1	1393/05/09
1	5	پاسخ 1 به پاسخ 4	1393/05/09
3	6	پاسخ 3 به ریشه 1	1393/05/09
1	7	پاسخ 1 به پاسخ 6	1393/05/09
4	8	پاسخ 4 به ریشه 1	1393/05/09
1	9	پاسخ 1 به پاسخ 8	1393/05/09
5	10	پاسخ 5 به ریشه 1	1393/05/09
1	11	پاسخ 1 به پاسخ 10	1393/05/09
2	12	نظر ریشه 12	1393/05/09
3	23	نظر ریشه 23	1393/05/09
4	34	نظر ریشه 34	1393/05/09
5	45	نظر ریشه 45	1393/05/09
6	56	نظر ریشه 56	1393/05/09
7	67	نظر ریشه 67	1393/05/09

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 treeGridModel حالات Nested Set model و Adjacency model قابل تنظیم هستند و ExpandColumn نام ستونی را مشخص می‌کند که قرار است فرزندان آن نمایش داده شوند.

یک نکته‌ی تکمیلی

اگر می‌خواهید دقیقا به شکل زیر برسید:

نظر
نظر ریشه 1
پاسخ 1 به ریشه 1
پاسخ 1 به پاسخ 2
پاسخ 2 به ریشه 1
پاسخ 1 به پاسخ 4
پاسخ 3 به ریشه 1
پاسخ 1 به پاسخ 6
پاسخ 4 به ریشه 1
پاسخ 1 به پاسخ 8
پاسخ 5 به ریشه 1
پاسخ 1 به پاسخ 10

تنظیم rownumbers: true گرید را حذف کنید. همچنین ستون Id را نیز با تنظیم‌های hidden:true, key: true مخفی نمائید (در تعاریف colModel).

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 می‌شود inline edit انجام داد؟
در صفحه‌ی ویکی پلاگین، این خط در قسمت محدودیت‌های adjacency نوشته شده و سپس خط خورده.

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:
Functionality -> Add tree node (مثال رسمی)
Insert a row (روی صفحه کلیک راست کرده و سورس آن‌را مطالعه کنید)
Add new rows to jqGrid Treegrid model (در استک اورفلو جستجو کنید، مطالب خوبی در مورد jqGrid دارد. تا امروز 8000 سؤال 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 BlogComment مطلب فوق)، با توضیحات فوق سازگار خواهد بود. اگر مستقیما SQL می‌نویسید، در مطلب « SQL Antipattern #2 » کوئری‌های آن موجود است. اگر با LINQ و EF کار می‌کنید، توضیحات مطلب «مباحث تکمیلی مدل‌های خود ارجاع دهنده در EF Code first » را پیگیری کنید.

+ 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 امضای متد فوق که در بحث مطرح شده، node id تعیین کننده‌ی واکشی 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 شده‌است. در اینجا فیلد parent id می‌تواند به عنوان کلید خارجی که به جدولی دیگر اشاره می‌کند نیز تفسیر 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 نهایتا 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: ۱۳۹۳/۰۹/۱۰ ۱۶:۵۱

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: ۱۳۹۳/۰۹/۱۰ ۱۸:۲۴

- 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.

عنوان: **سازگارسازی کلاس‌های اعتبارسنجی Twitter Bootstrap 3 با فرم‌های ASP.NET MVC**
نویسنده: علی اکبر کورش فر
تاریخ: ۱۳۹۳/۰۵/۱۰ ۱۹:۱۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, Twitter Bootstrap

چندی پیش در همین وب‌سایت مطلبی تحت عنوان « اعمال کلاس‌های ویژه اعتبارسنجی Twitter bootstrap به فرم‌های ASP.NET MVC » منتشر شد. این مقاله مرتبط با نسخه دوم فریم‌ورک محبوب Bootstrap بود. قصد داریم به بازنویسی کدهای مرتبط بپردازیم و کلاس‌های مرتبط با نسخه سوم این 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} یک فیلد اجباری است و باید آن را وارد کنید.")]
        [StringLength(50, ErrorMessage = "طول {0} باید کمتر از {1} کاراکتر باشد.")]
        [Display(Name = "نام کالا")]
        public string Name { get; set; }
        [Required(ErrorMessage = "{0} یک فیلد اجباری است و باید آن را وارد کنید.")]
        [Display(Name = "قیمت")]
        [DataType(DataType.Currency)]
        public double Price { get; set; }
        [Required(ErrorMessage = "{0} یک فیلد اجباری است و 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 مد نظر 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", "فقط محصولی با 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 گفتیم اگر نام محصول چیزی غیر از «پفک» بود، از سمت سرور خطایی را صادر کند و بگوید که فقط پفک قابل ثبت هست.

View برنامه

```
@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>
                @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>
                @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>
                @Html.EditorFor(model => model.Qty, new { htmlAttributes = new { @class = "form-
control" } })
                @Html.ValidationMessageFor(model => model.Qty, "", new { @class = "text-danger" })
            </div>
        </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 Templateها ساخته شده و چون از Visual Studio 2013 استفاده شده، به‌صورت پیش‌فرض با بوت‌استرپ سازگار هست. تغییری که ایجاد شده تعویض کلاس مربوط به ValidationSummary هست که به alert alert-danger تغییر پیدا کرده و همچین دو دکمه «ریست» و «بازگشت به لیست» هم به کنار دکمه «ثبت» اضافه شده.

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 Bootstrap باشند حاصل نشده. برای رفع این مشکل کافی‌است 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 و 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:

همچنین هنگامیکه کاربر فیلد را به درستی وارد کرد، رنگ فیلد و همچین آن ردیف به سبز تغییر خواهد کرد.

و همچنین در حالت رخداد یک خطای سفارشی پس از postback از سمت سرور به حالت زیر خواهیم رسید.

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید

FormValidationWithBootstrap.rar

نظرات خوانندگان

نویسنده: مهدی پایروند
تاریخ: ۱۳۹۳/۰۵/۱۰ ۲۲:۳۴

سلام لطفا مثال پست را هم برای دانلود بذارید. ممنون

نویسنده: علی اکبر کورش فر
تاریخ: ۱۳۹۳/۰۵/۱۱ ۱۳:۳۳

سلام. متاسفانه حجم پروژه به دلیل فایل‌های دی‌ال‌ال موجود 18 مگ بود و من نتونستم کمترش کنم

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۱۱ ۱۳:۳۸

برای ارسال پروژه‌ها در وب، همیشه پوشه‌های **packages** ، پوشه‌های **bin** و **obj** را حذف کنید. نیازی به آن‌ها نیست، چون قابل بازسازی هستند.

نویسنده: علی اکبر کورش فر
تاریخ: ۱۳۹۳/۰۵/۱۱ ۲۳:۰

خیلی ممنونم. خیلی دنبال این مسئله بودم. مچکرم از راهنماییتون

نویسنده: علی اکبر کورش فر
تاریخ: ۱۳۹۳/۰۵/۱۱ ۲۳:۱

با عرض سلام. پروژه پیوست شد

نویسنده: هادی ام
تاریخ: ۱۳۹۳/۰۵/۱۳ ۱۹:۲۱

این بوتسترپ شما، فارسی شده هست؟

نویسنده: علی اکبر کورش فر
تاریخ: ۱۳۹۳/۰۵/۱۴ ۸:۲۸

فارسی‌شده در حد نیاز... قسمت‌هایی که ازشون استفاده می‌کنم رو فارسی کردم.
فکر می‌کنم نسخه فارسی سازی شده بوت‌استرپ 3 هم هستش

نویسنده: محمد دلیری
تاریخ: ۱۳۹۳/۰۵/۲۰ ۰:۵۱

اون قسمت‌های که میگی کجاها واسه فارسی کردن تغییر دادین که راست به چپ شده ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۲۰ ۱:۳۲

- نسخه 3 راست به چپ
- نسخه 3.1.1 راست به چپ
- نسخه 3.2 راست به چپ

عنوان: **اعتبارسنجی در فرم‌های ASP.NET MVC با Remote Validation**
نویسنده: علی اکبر کورش فر
تاریخ: ۱۳۹۳/۰۵/۱۱ ۱۲:۵۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, Validation, Attributes

بعد از آمدن نسخه‌ی سوم ASP.NET MVC مکانیسمی 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، سریعا مشخص می‌شود که آیا این نام کاربری قابل استفاده برای شما هست یا خیر.
به‌صورت معمول برای انجام این کار باید با جاوا اسکریپت، مدیریتی روی فیلد مربوطه انجام دهیم. مثلا 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 قبلا گرفته شده یا نه.
انجام این کار به‌راحتی با مزین‌کردن 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 چک‌کردن 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} یک فیلد 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} یک فیلد اجباری است و باید آن را وارد کنید.")]
        [DataType(DataType.Currency)]
        public double Price { get; set; }

        [Display(Name = "بارکد")]
        [Required(ErrorMessage = "{0} یک فیلد 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 چک‌کردن از مهم‌ترین قسمت‌های اصلی هست. چیزهایی دیگه‌ای هم هست که می‌توانیم آن‌ها را مقداردهی کنیم. مثل HttpMethod، ErrorMessage و یا AdditionFields. HttpMethod 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 به‌همراه فیلد مربوطه به سمت سرور بفرستیم. مثلا می‌تونیم به‌همراه 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 چک‌کردن مقدارهای لازم و در پایان آن یک خروجی Json را برمی‌گردانیم که مقدار true یا false دارد. در حقیقت مقدار را به این صورت برمی‌گردانیم که اگر مقدار ورودی در پایگاه داده وجود دارد، false را برمی‌گرداند و اگر وجود نداشت true. همین‌طور آمدیم از کش شدن درخواست‌هایی که با Ajax آمده با ویژگی OutputCache جلوگیری کردیم.

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 می‌خواد 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 برگرده. این فقط در حالتی هستش که کاربر بخواد مقدار رو تغییر بده. 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 روبرو میشویم این است که هنگام ویرایش یک رکورد موجود در بانک ، اگر قرار نباشد فیلد مورد نظر بروز رسانی شود ، اکشن متد ذکر 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. مثلا 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 روال‌های رخدادگردان عمومی ajaxStart و ajaxComplete استفاده کنید.

نویسنده: م علی خسروی
تاریخ: ۱۳۹۳/۰۷/۲۳ ۹:۷

با تشکر
در واقع من سه input دارم که می‌خام remote بشند 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 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، متد startRequest پیش از شروع عملیات Ajax و متد

stopRequest پس از پایان کار فراخوانی می‌شوند.

```
prototype: {
startRequest: function( element ) {
  //...
},
stopRequest: function( element, valid ) {
  //...
},
```

این دو متد را باید برای نمایش loading بازنویسی کرد. برای مثال:

```
var originalStartRequest = $.validator.prototype.startRequest;
$.validator.prototype.startRequest = function (element) {
    // یافتن عنصر در حال بررسی
    var container = $('form').find("[data-valmsg-for='" + element.name + "']");
// افزودن کلاس نمایش منتظر بمانید
    container.addClass('loading');

// فراخوانی متد اصلی برای انجام کارهای درونی افزونه
    originalStartRequest.apply(this, arguments);
};

var originalStopRequest = $.validator.prototype.stopRequest;
$.validator.prototype.stopRequest = function (element) {
    // یافتن عنصر در حال بررسی
    var container = $('form').find("[data-valmsg-for='" + element.name + "']");
// حذف  کلاس نمایش منتظر بمانید
    container.removeClass('loading');

// فراخوانی متد اصلی برای انجام کارهای درونی افزونه
    originalStopRequest.apply(this, arguments);
};
```

در اینجا loading به span مخفی data-valmsg-for اضافه می‌شود.

```
<span class="field-validation-valid" data-valmsg-replace="true" data-valmsg-for="Url"></span>
```

نمونه‌ی این بازنویسی در مطلب « اعتبار سنجی سمت کاربر wysiwyg-editorها در ASP.NET MVC » هم انجام شده‌است.

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 دیباگ برنامه‌های ASP.NET مبتنی بر jQuery

عنوان: قالب‌های سفارشی برای HtmlHelperها
نویسنده: علی اکبر کورش فر
تاریخ: ۱۳۹۳/۰۵/۱۶ ۱۳:۳۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, HTML helpers

در ابتدای بحث، برای آشنایی بیشتر با HTML Helperها به مطالعه این مقاله بپردازین.
در این مقاله قرار است برای یک HTML Helper خاص، قالب نمایشی اختصاصی خودمان را طراحی کنیم و به نحوی HTML Helper 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 اصلا 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>
```

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 بتونیم بعدا 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 نشده‌است. لیست کتاب‌ها را از پایگاه داده بیرون آوردیم و از طریق اکشن Index به نمایش گذاشتیم. با اکشن‌های Edit، Create 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.

View مربوط به اکشن Create برنامه

```
@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")
}
```

View برنامه هم همان ویویی است که خود Visual Studio برای ما ساخته‌است. به جز یک سری دست‌کاری‌هایی داخل سی‌اس‌اس، هدف از گذاشتن View مربوط به Create این بود که قرار است بر روی این قسمت کار کنیم. اگر پروژه رو اجرا کنید و به قسمت Create بروید، مشاهده خواهید کرد که برای Genre یک input ساخته شده‌است 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 عملا باید یک لیست به کاربر نشان داده شود که تا از آن لیست، نوع را انتخاب کند. می‌توانیم بیایم همینجا در داخل View مربوطه، به‌جای استفاده از HTML Helper پیش‌فرض، از DropDownList یا EnumFor استفاده کنیم و به طریقی این لیست را ایجاد کنیم. ولی چون قرار است در این مثال به شرح موضوع مقاله خودمان بپردازیم، این کار را انجام نمی‌دهیم.

در حقیقت می‌خوایم متد EditorFor را طوری سفارشی سازی کنیم که برای نوع شمارشی Genre، به صورت خودکار یک لیست ایجاد کرده و برگرداند. از نسخه‌ی سوم ASP.NET MVC به بعد این امکان برای توسعه دهنده‌ها فراهم شده‌است. شما می‌توانید در پوشه‌ی Shared داخل پوشه Views برنامه، پوشه‌ای را به اسم EditorTemplates ایجاد کنید؛ همینطور DisplayTemplates و برای نوع خاصی که می‌خواهید سفارشی‌سازی را برای آن انجام دهید، یک PartialView بسازید.

```
Views/Shared/DisplayTemplates/<type>.cshtml
```

یک PartialView در داخل پوشه EditorTemplates به نام Genre.cshtml ایجاد کنید. برای اینکه مشاهده کنید چطور کار می‌کند، کافی‌است یک فایل متنی اینجا تهیه کرده و بعد پروژه را اجرا کرده و به قسمت Create روید تا تغییرات را مشاهده کنید. بله! به‌جای inputی که از قبل وجود داشت، فقط متن شما آنجا نوشته شده‌است. (به عکسی که در بالا قرار دارد هم می‌تونید نگاه کنید)

کاری که الان میخواهیم انجام دهیم این است که یک SelectListItem ایجاد کرده تا مقدارهای نوع Genreمان داخلش باشد و بتوانیم به راحتی برای ساختن DropDownList از آن استفاده کنیم. برای این کار Helper مخصوص خودمان را ایجاد می‌کنیم. پوشه‌ای به اسم Helpers در کنار پوشه‌های 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 به‌صورت متدهای الحاقی به نوع Type نوشته شدند. کار خاصی در بدنه‌ی متدها انجام نشده‌است. در بدنه‌ی متد اول لیست آیتم‌ها را تولید کردیم. در هنگام ساخت SelectListItem برای گرفتن Text، متد GetName را صدا زدیم. برای اینکه بتوانیم مقدار ویژگی Display که در هنگام تعریف نوع شمارشی استفاده کردیم را بدست بیاریم، باید چک کنیم ببینیم که آیا این آیتم به این ویژگی مزین شده‌است یا نه. اگر شده بود مقدار را می‌گیریم و به خصوصیت Text متد 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 PartialViewی Genre قرار دهیم که در پوشه‌ی 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.

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید

HtmlHelpersEditor.rar

عنوان: **ارسال ویدیو بصورت Async توسط Web Api**
نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۳/۰۵/۲۶ ۱۵:۱۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, ASP.NET Web API, Asynchronous Programming, ASP.NET Web API 2, Network Programming

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 *WebApiConfig.cs* در پوشه *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 بر می‌گرداند.

متد سازنده این کلاس پارامتری از نوع رشته بنام *videoPath* را می‌پذیرد که مسیر کامل فایل ویدیوی 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 پروسه‌های 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 (مجددا 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 (مثلا 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. مثلا 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 کنترلرها 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 ext و fileName 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 نهایتا همین کلاس HostingEnvironment 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 وهله جدیدی از کلاس VideoStream می‌سازیم و مسیر کامل فایل را به آن پاس می‌دهیم.

```
var video = new VideoStream(videoPath);
```

سپس آبجکت پاسخ را وهله 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 Headerها را تعریف نکنید سایز فایل دریافتی و مدت زمان آن نامعلوم خواهد بود که تجربه کاربری خوبی بدست نمی‌دهد. نهایتا 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 bravity
}
else
{
    return Request.CreateResponse(HttpStatusCode.NotFound);
}
```

خوب، برای تست این مکانیزم نیاز به یک کنترلر MVC و یک View داریم. در پوشه کنترلرها 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 المنت 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 المنت 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 پکیج‌های 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 المنت *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 حجیم‌تر 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 کجای فیلم از سرور دانلود شده (در صورتی که در حالت پخش مستقیم ویدیو از لینک مستقیم اینگونه نیست).

ممنون میشم اگر به این سه سوال پاسخ بدین :
1- مزیت این روش نسبت به روشی که از لینک مستقیم فایل ویدیو استفاد 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

نویسنده: رضا بهشتی
تاریخ: ۱۶:۴ ۱۳۹۴/۰۱/۲۶

بسیار عالی بود، شما میدونید چجوری میشود بجای یک فایل ویدویی از یک منبع مثلا وب کم استفاده کرد؟

نویسنده: محسن خان
تاریخ: ۱۶:۴۴ ۱۳۹۴/۰۱/۲۶

IIS یک سری ابزار برای اینکار داره:

IIS Media Services: http://www.iis.net/media
Smooth Streaming Client: http://www.iis.net/download/SmoothClient

همچنین اصل کار به DirectX ختم میشه. دو نمونه پیاده سازی:

http://www.codeproject.com/Articles/7637/DirectX-Video-Stream-and-frame-capture

/http://channel9.msdn.com/forums/TechOff/93476-Programatically-Using-A-Webcam-In-C

عنوان: اعتبار سنجی سمت کاربر wysiwyg-editor ها در ASP.NET MVC
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۶/۰۵ ۱۲:۵۰
آدرس: www.dotnettips.info
گروه‌ها: MVC

تفاوتی نمی‌کند که از کدامیک از HTML Editorها یا به عبارتی wysiwyg-editorهای موجود، جهت ورود اطلاعات استفاده می‌کنید. هیچکدام از آن‌ها سبب فراخوانی اعتبارسنجی Required سمت کاربر نمی‌شوند. چرا؟
علت اینجا است که با فعال سازی wysiwyg-editorهای موجود، المان 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 آن‌را به "" تنظیم کنید.

مشکل دوم! متد required پیش فرض، فقط به رشته‌های خالی یا نال واکنش نشان می‌دهد. اما اگر کاربری در اینجا

**
 را وارد کرد، چطور؟**

در این حالت نیاز است متد rqeuired 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 ایی را حذف می‌کند. متد trim جی‌کوئری نیز سبب حذف فواصل خالی در ابتدا 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 فراخوانی گردد. در ادامه با فراخوانی jQuery.validator.addMethod و بکارگیری نام required، دقیقا 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 jQuery.validator.messages.required اصل پیام خطای تنظیم شده به کاربر نمایش داده خواهد شد.

نظرات خوانندگان

نویسنده: آرمین ضیاء
تاریخ: ۱۳۹۳/۰۶/۰۵ ۲۱:۱۵

مرسی بابت این پست، راه حل جدید و کارآمدی یاد گرفتم.
پیش از این برای ادیتورهای Froala به نحو زیر عمل می‌کردم:

```
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 خطا Focus کرد و یا کلاس CSS خاصی بدان اضافه کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۶/۰۵ ۲۱:۵۵

در همان callback اعتبارسنجی (function (value, element, param به element 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 گردان می‌توان نوشت:

```
$(".froala-element").focus();
$(".froala-element").css({"border-color": "red"});
```

نویسنده: محمدرضایا
تاریخ: ۱۳۹۳/۰۶/۲۶ ۱۰:۱

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 کارکتر دیگه میتونه وارد کنه و تاحالا چندتا زده ولی روی WYSIWYG Editor جواب نداد!
من از froala editor استفاده کرده ام.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۶/۲۶ ۱۰:۱۶

- در مورد پروژه‌های سایت‌های ثالث، به issue tracker آن‌ها مراجعه کنید و سپس درخواست دهید تا متد removeAllTagsAndTrim فوق را هم تحت عنوان یک گزینه‌ی جدید اضافه کنند.
- در مورد موضوع بحث جاری، در فایل jquery.validate.js سه متد maxlength ، minlength و rangelength برای بررسی طول حروف وارد شده، مورد استفاده قرار می‌گیرند. بازنویسی آن‌ها جهت اعمال متد removeAllTagsAndTrim، دقیقا شبیه به بازنویسی متد required است که در بحث جاری ملاحظه کردید.

نویسنده: محمدرضایا
تاریخ: ۱۳۹۳/۰۶/۲۶ ۱۰:۴۷

متد maxlength را چطور باید باز نویسی کنم؟
راه حلی که در متن ذکر کردید قبلا رو Editor اعمال کرده بودم حالا باید در ادامه اون تکه کد متد maxlength را بنویسم و

همچنین نیاز به Attribut حداکثر تعداد کاراکتر maxlength مربوط به تگ textarea هم هست؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۶/۲۶ ۱۱:۴

کلیات آن تفاوتی نمی‌کند. فقط در کدهای فوق، هر جایی required هست، آن‌را تبدیل کنید به maxlength. چون در پایان کار، متد اصلی maxlength فراخوانی می‌شود، روش بکار گرفته شده، عمومی است و قابل تعمیم به تمام متدهای فایل jquery.validate.js .

نویسنده: امیر نوروزیان
تاریخ: ۱۳۹۳/۱۰/۲۱ ۲۲:۴۲

با سلام؛ تو سورس کد من چنین کدی رو نمی‌بینیم:

```
$.extend( $.validator, {
defaults: {
ignore: ":hidden" ,
```

کدهای جاری را دقیقا کجا بزارم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۲۱ ۲۲:۵۳

- این تنظیم پیش فرض در فایل jquery.validate.js موجود است (عبارت ":hidden" را در آن جستجو کنید).
- فقط متد jQuery.validator.setDefaults یاد شده را به همراه پارامترهایی که ذکر شد، یکبار در صفحه فراخوانی کنید (پس از الحاق فایل jquery.validate.js).

عنوان: **بررسی مقدمات کتابخانه‌ی JSON.NET**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۶/۱۳ ۱۹:۰
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, MVC, JSON, ASP.NET Web API, json.net

چرا JSON.NET؟

JSON.NET یک کتابخانه‌ی سورس باز کار با اشیاء JSON در دات نت است. تاریخچه‌ی آن به 8 سال قبل بر می‌گردد و توسط یک برنامه نویس نیوزیلندی به نام James Newton King تهیه شده‌است. اولین نگارش آن در سال 2006 ارائه شد؛ مقارن با زمانی که اولین استاندارد JSON نیز ارائه گردید.

این کتابخانه از آن زمان تا کنون، 6 میلیون بار دانلود شده‌است 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 مایکروسافت 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 2 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

کتابخانه‌ی JSON.NET از سه قسمت عمده تشکیل شده‌است:

الف) JsonSerializer

ب) LINQ to JSON

ج) JSON Schema

الف) JsonSerializer

کار JsonSerializer تبدیل اشیاء دات نتی به JSON و برعکس 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، پیش بینی شده‌است.

یک مثال:

```
var roles = new List<string>
{
   "Admin",
   "User"
};
string json = JsonConvert.SerializeObject(roles, Formatting.Indented);
```

در اینجا نحوه‌ی استفاده از JSON.NET را جهت تبدیل یک شیء دات نتی، به معادل JSON آن مشاهده می‌کنید. اعمال تنظیم Formatting.Indented سبب خواهد شد تا خروجی آن دارای Indentation باشد. برای نمونه اگر در برنامه‌ی خود قصد دارید فرمت JSON تو در تویی 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 نحوه‌ی معرفی JsonSerializerSettings را مشاهده می‌نمائید:

```
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، می‌توان خروجی DateTime 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 متد JsonConvert.SerializeObject یک رشته‌است که در صورت نیاز به سادگی توسط متد 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 JsonSerializer و متد Serialize آن یک استریم را نیز جهت نوشتن خروجی می‌پذیرند. برای مثال response.Output 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 نهایتا GC کار کمتری را باید انجام دهد.

تبدیل JSON رشته‌ای به اشیاء دات نت

اگر رشته‌ی jsonData ایی 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 متد DeserializeObject به همراه مشخص سازی صریح نوع شیء نهایی استفاده شده‌است.
البته در اینجا با توجه به استفاده از JavaScriptDateTimeConverter برای تولید jsonData، 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 وهله‌ی از پیش موجود

متد JsonConvert.DeserializeObject یک شیء جدید را ایجاد می‌کند. اگر قصد دارید صرفا تعدادی از خواص یک وهله‌ی موجود، توسط JSON.NET مقدار دهی شوند از متد PopulateObject استفاده کنید:

```
JsonConvert.PopulateObject(jsonData, user);
```

کاهش حجم JSON تولیدی

زمانیکه از متد JsonConvert.SerializeObject استفاده می‌کنیم، تمام خواص عمومی تبدیل به معادل JSON آن‌ها خواهند 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 مشخص گردد.

```
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. مثلا 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 ایی 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 وهله از User1 خواهیم رسید و نه یک وهله‌ی 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 متد 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 وهله متفاوت از شیء با Id=2 تولید نشود.

باید دقت داشت که در حین استفاده از JsonConvert.DeserializeObject نیز باید JsonSerializerSettings یاد شده، تنظیم شوند.

ویژگی‌های قابل تنظیم در JSON.NET

علاوه بر JsonSerializerSettings که از آن صحبت شد، در JSON.NET امکان تنظیم یک سری از ویژگی‌ها به ازای خواص مختلف نیز وجود دارند.
- 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; }
}
```

JsonIgnore 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 TypeNameHandling، NullValueHandling 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 صریحا 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 Y رندر شود.

- استفاده از ویژگی JsonObject به همراه مقدار OptIn آن به این معنا است که از کلیه خواصی که دارای ویژگی JsonProperty نیستند، صرفنظر شود. حالت پیش فرض آن OptOut است؛ یعنی تمام خواص عمومی در خروجی JSON حضور خواهند 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 JsonConverterها می‌توان کنترل کاملی را بر روی اعمال serialization و deserialization 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 علاقمندیم، 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 صرفا قرار است برای حالت serialization استفاده شود، قسمت ReadJson آن پیاده سازی نشده‌است.

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

نویسنده: افتابی
تاریخ: ۱۳۹۳/۰۶/۱۳ ۲۱:۵۴

سلام؛ من مطالب مربوطه رو خوندم فقط اینکه توی یه صفحه rozar در mvc من به چه نحو میتونم از آن استفاده کنم ، حتی توی سایت خودش هم رفتم و sample ها رو دیدم فقط میخوام در یک پروژه به چه نحو ازش استفاده کنم و کجا کارش ببرم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۶/۱۳ ۲۱:۵۹

در یک اکشن متد، بجای return Json پیش فرض و توکار، می‌شود نوشت:

```
return Content(JsonConvert.SerializeObject(obj));
```

البته این ساده‌ترین روش استفاده از آن است؛ برای مقاصد Ajax ایی.
و یا برای ذکر Content type می‌توان به صورت زیر عمل کرد:

```
return new ContentResult
{
  Content = JsonConvert.SerializeObject(obj),
  ContentType = "application/json"
};
```

نویسنده: رحمت اله رضایی
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؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۶/۱۴ ۱۴:۱۹

- تا ASP.NET MVC 5 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 متداول 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 دقیقا از JSON.NET استفاده می‌کند.

نویسنده: 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);
                    }
                }
            }
        }
```

عنوان: **تنظیمات و نکات کاربردی کتابخانه‌ی JSON.NET**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۶/۱۴ ۱۱:۳۵
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, MVC, JSON, ASP.NET Web API, json.net, Entity framework

پس از بررسی مقدماتی امکانات کتابخانه‌ی JSON.NET، در ادامه به تعدادی از تنظیمات کاربردی آن با ذکر مثال‌هایی 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 تک خواص و مشخص کردن نام‌های مورد نیاز کتابخانه‌ی جاوا اسکریپتی به صورت صریح است.
روش دوم، استفاده از تنظیمات ContractResolver می‌باشد که با تنظیم آن به CamelCasePropertyNamesContractResolver به 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، دقیقا 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 درنظر 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 با 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]'.
```

اصل خطای معروف فوق «Self referencing loop detected» است. در اینجا کلاس‌هایی که به یکدیگر ارجاع می‌دهند، در حین عملیات Serialization سبب بروز یک حلقه‌ی بازگشتی بی‌نهایت شده و در آخر، برنامه با خطای stack overflow خاتمه می‌یابد.

راه حل اول:

به تنظیمات JSON.NET، مقدار ReferenceLoopHandling = ReferenceLoopHandling.Ignore را اضافه کنید تا از حلقه‌ی بازگشتی بی‌پایان جلوگیری شود:

```
return JsonConvert.SerializeObject(category, new JsonSerializerSettings
{
   Formatting = Formatting.Indented,
   ReferenceLoopHandling = ReferenceLoopHandling.Ignore,
   Converters = { new StringEnumConverter() }
});
```

راه حل دوم:

به تنظیمات JSON.NET، مقدار PreserveReferencesHandling = PreserveReferencesHandling.Objects را اضافه کنید تا مدیریت ارجاعات 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 شده‌است تا توسط آن بتواند ارجاعات و نمونه‌های اشیاء را تشخیص دهد.

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۶/۲۳ ۱۵:۱۷

گرفتن خروجی مرتب شده بر اساس نام خواص (جهت مقاصد نمایشی):

تعریف 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()
            });
```

عنوان: **استفاده از JSON.NET در ASP.NET MVC**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۶/۱۵ ۲۳:۲۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, JSON, json.net

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 کندتر 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 متد، 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 JsonConvert.SerializeObject، اطلاعات شیء مدنظر تبدیل به یک رشته شده و سپس با content type 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، مستقیما 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 ارث بری از JsonResult توکار ASP.NET MVC شروع شده‌است. کدهای ابتدای متد ExecuteResult با کدهای اصلی JsonResult یکی هستند. فقط انتهای کار بجای استفاده از JavaScriptSerializer، از JSON.NET استفاده شده‌است.
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 پیشرفته‌ای نیز به آن اضافه شده‌است.

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 دقت کنید از کلاس JavaScriptSerializer استفاده شده‌است.
برای جایگزینی آن باید یک ValueProvider جدید را تهیه کنیم:

```
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 JsonNetValueProviderFactory : 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();
```

```
                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 پردازش نخواهند شد.

در ادامه، اطلاعات JSON دریافتی به شکل یک رشته‌ی خام دریافت شده و سپس به متد JsonConvert.DeserializeObject ارسال می‌شود. با استفاده از تنظیم ExpandoObjectConverter، می‌توان محدودیت کلاس JavaScriptSerializer را در مورد خواص و یا پارامترهای dynamic، برطرف کرد.

```
[HttpPost]
  public ActionResult TestValueProvider(string data1, dynamic data2)
```

برای مثال اینبار می‌توان اطلاعات دریافتی را همانند امضای متد فوق، به یک پارامتر از نوع dynamic، بدون مشکل نگاشت کرد.

و 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 JsonNetValueProviderFactory());
        }
    }
}
```

ابتدا نمونه‌ی قدیمی آن یعنی JsonValueProviderFactory حذف می‌شود و سپس نمونه‌ی جدیدی که از JSON.NET استفاده می‌کند، معرفی خواهد شد.

البته نگارش بعدی ASP.NET MVC موتور پردازشی JSON خود را از طریق تزریق وابستگی‌ها دریافت می‌کند و از همان ابتدای کار قابل تنظیم و تعویض است. مقدار پیش فرض آن نیز به JSON.NET تنظیم شده‌است.

دریافت یک مثال کامل

MvcJsonNetTests.zip

نظرات خوانندگان

نویسنده: مهدی پایروند
تاریخ: ۱۳۹۳/۰۶/۱۶ ۱۱:۱۸

تا جایی که من میدونم JavaScriptSerializer با Dictionaryها هم مشکل داشت ولی JSON.NET این مشکل رو نداره.

نویسنده: احمد
تاریخ: ۱۳۹۳/۰۶/۱۶ ۱۳:۲۲

سلام. اگر در مثال پیوست شده کلاس زیر را استفاده کنیم خطا می‌دهد:

```
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 JsonNetResult
            {
                Data = new { result = data1 },
                JsonRequestBehavior = JsonRequestBehavior.AllowGet,
                Settings = { ReferenceLoopHandling = ReferenceLoopHandling.Ignore }
            };
        }
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۶/۱۶ ۱۵:۱۷

نسخه‌ی بهبود یافته JsonNetValueProviderFactory را در اینجا می‌توانید مطالعه کنید. نسخه‌ی JsonNetResult آن جالب نیست چون از string استفاده کرده بجای stream .

JsonNetValueProviderFactory.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 JsonNetValueProviderFactory());
        }
```

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)
{
```

Enum 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' // مقدار دهی عضو ایی‌نام
                }),
```

عنوان: LINQ to JSON به کمک JSON.NET
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۶/۱۹ ۲۳:۵۵
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, MVC, JSON, ASP.NET Web API, json.net

عموما از امکانات LINQ to JSON کتابخانه‌ی JSON.NET زمانی استفاده می‌شود که ورودی JSON تو در توی حجیمی را دریافت کرده‌اید اما قصد ندارید به ازای تمام موجودیت‌های آن یک کلاس معادل را جهت نگاشت به آن‌ها تهیه کنید و صرفا یک یا چند مقدار تو در توی آن جهت عملیات استخراج نهایی مدنظر است. به علاوه در اینجا LINQ to JSON واژه‌ی کلیدی dynamic را نیز پشتیبانی می‌کند.

همانطور که در تصویر مشخص است، خروجی‌های JSON عموما ترکیبی هستند از مقادیر، آرایه‌ها و اشیاء. هر کدام از این‌ها در LINQ to JSON به اشیاء JArray، JValue و JObject نگاشت می‌شوند. البته در حالت JObject هر عضو به یک JProperty و JValue تجزیه خواهد شد.

برای مثال آرایه [1,2] تشکیل شده‌است از یک JArray به همراه دو JValue که مقادیر آن‌را تشکیل می‌دهند. اگر مستقیما 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 دریافتی را داریم می‌توان از متد 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 JTokenها است و با آن می‌توان مانند لیست‌های معمولی کار کرد.

در حالت کار با اشیاء، شیء JObject امکان تهیه اشیاء JSON ایی را دارا است که می‌تواند مجموعه‌ای از JPropertyها باشد:

```
var jObject = new JObject(
new JProperty("prop1", "value1"),
new JProperty("prop2", "value2")
);
var jObjectToJson = jObject.ToString();
```

با JObject به صورت dynamic نیز می‌توان کار کرد:

```
dynamic jObj = new JObject();
jObj.Prop1 = "value1";
jObj.Prop2 = "value2";
jObj.Roles = new[] {"Admin", "User"};
```

این روش بسیار شبیه است به حالتی 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 JObject از نوع IEnumerable است و بر روی آن‌ها می‌توان کلیه متدهای LINQ را فراخوانی کرد. برای مثال در اینجا اولین شیءایی 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 شده‌است:

```
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.

عنوان: **آشنایی با چالش های امنیتی در توسعه برنامه‌های تحت وب، بخش اول**
نویسنده: سید مهران موسوی
تاریخ: ۱۳۹۳/۰۷/۱۱ ۸:۵۵
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, MVC, Security, xss, Cross-site scripting

در پروژه‌های بزرگ نرم افزاری، از قدیم بحث تامین امنیت پروژه، یکی از چالش‌های مهم بوده است. از دیدگاه شخصی بنده، یک مدیر نرم افزار یا حتی یک توسعه دهنده‌ی برنامه‌های تحت وب، لازم است علاوه بر صرف وقت مطالعاتی و آشنایی و تسلط بر مباحث طراحی معماری سیستم‌های تحت وب، که از اهمیت بالا و مقیاس بزرگی برخوردارند آشنایی لازم را با چالش‌های امنیتی در پیاده سازی اینگونه سیستم‌ها داشته باشد. امنیت در یک سیستم بزرگ و ارائه دهنده خدمات، باعث می‌شود تا کاربر علاوه بر یک تجربه کاربری (user experience) خوب از سیستم که حاصل پیاده سازی صحیح سیستم می‌باشد، اعتماد ویژه‌ای به سیستم مذکور داشته باشد. گاها کاربران به علت بی اعتمادی به شرایط امنیتی حاکم بر یک سیستم، از تجربه کاربری خوب یک سیستم چشم پوشی می‌کنند. اهمیت این مسئله تا جاییست که غول‌های تکنولوژی دنیا همچون Google درگیر این چالش می‌باشند و همیشه سعی بر تامین امنیت کاربران علاوه بر ایجاد تجربه کاربری خوب دارند. پس عدم توجه به این موضوع میتواند خسارات وارده جبران ناپذیری را به یک سیستم از جهت‌های مختلف وارد کند.

در این سری از مقالات، بنده سعی دارم تا حد توان در رابطه با چالش‌های امنیتی موجود در زمینه توسعه برنامه‌های 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 چالش‌های امنیتی زیر مورد بحث و بررسی واقع خواهند 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** میپردازیم .

واژه XSS مخفف Cross-site scripting ، نوعی از آسیب پذیریست که در برنامه‌های تحت وب نمود پیدا میکند. به طور کلی و خلاصه، این آسیب پذیری به فرد نفوذ کننده اجازه تزریق اسکریپت‌هایی را به صفحات وب، می‌دهد که در سمت کاربر اجرا می‌شوند (Client Side scripts) . در نهایت این اسکریپت‌ها توسط سایر افرادی که از صفحات مورد هدف قرار گرفته بازدید می‌کنند اجرا خواهد شد.

هدف از این نوع حمله :

بدست آوردن اطلاعات کوکی‌ها و سشن‌های کاربران (مرتبط با آدرسی که صفحه آلوده شده در آن قرار دارد) است. سپس فرد نفوذ کننده متناسب با اطلاعات بدست آمده می‌تواند به اکانت شخصی کاربران مورد هدف قرار گرفته، نفوذ کرده و از اطلاعات شخصی آن‌ها 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 میشوند. محل ذخیره سازی می‌تواند دیتابیس سایت یا هر جای دیگری که داده‌ها توسط سایت یا برنامه تحت وب بازیابی می‌شوند و نمایش داده می‌شوند باشد. اما اینکه چگونه کدهای مخرب در منابع یاد شده ذخیره میشوند؟

فرض کنید در سایت جاری آسیب پذیری مذکور وجود دارد. راه‌های ارسال داده‌ها به این سایت چیست؟ نویسندگان میتوانند مطلب ارسال کنند و کاربران میتوانند نظر دهند. حال اگر در یکی از این دو بخش بررسی‌های لازم جهت مقابله با این آسیب پذیری

وجود نداشته باشد و نوشته‌های کاربران که می‌تواند شامل کدهای مخرب باشد مستقیما در دیتابیس ذخیره شده و بدون هیچ اعتبار سنجی نمایش داده شود چه اتفاقی رخ خواهد داد؟ مسلما 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 وبسایت یا اپلیکیشن تحت وب توسط فرد مهاجم ذخیره نمی‌شود ! بلکه از ضعف امنیتی بخش‌هایی همچون بخش جستجو وب سایت، بخش‌های نمایش پیغام خطا و ... استفاده میشود ... اما به چه صورت؟

در بسیاری از سایت‌ها، انجمن‌ها و سیستم‌های سازمانی تحت وب، مشاهده می‌شود که مثلا 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 بعضا 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=PHNjcmlwdD5hbGVydChkb2N1bWVudC5jb29raWUpOzwvc2NyaXB0Pg==
```

این آدرس حاصل submit شدن فرم جستجو وب‌سایت test (نام وب‌سایت 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 وبسایت 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 وبسایت 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 (مثلا 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 مرورگرش اجرا شده و کوکی‌های سایت مذکور برای مهاجم ارسال خواهد شد ... به این نوع حمله XSS ، نوع انعکاسی می‌گویند که کاملا 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 هکرها 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 حاظر این باگ در یاهو وجو 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 پذیری هایی از این دست در تمامی زبان‌ها و تکنولوژی‌های تحت وب یکسان میباشند.

خوشبختانه کتابخانه‌ای 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 می‌کند. 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 می‌شوند. این داستان برای Url ها هم که به صورت پیش فرض encode میشوند صدق می‌کند. ولی گاها وقتی شما برای ورود اطلاعات مثلا از یک ادیتور WYSWYG استفاده می‌کنید و نیاز دارید داده‌ها را بدون encoding رندر کنید. آنگاه به ناچار مجاب بر اعمال یک سری سیاست‌های خاص‌تر بر روی داده مورد نظر برای رندر می‌شوید و نمی‌توانید از encoding توکار فوق الذکر استفاده کنید. آنگاه این کتابخانه در اعمال سیاست‌های جلوگیری از بروز این آسیب پذیری می‌تواند برای شما مفید واقع شود.

```
PM> Install-Package AntiXSS
```

این کتابخانه مجموعه‌ای از توابع کد کردن عبارات است که از مواردی همچون Html, XML, Url, Form, LDAP, CSS, JScript and VBScript پشتیبانی می‌کند. استفاده از آن بسیار ساده می‌باشد. 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: ۱۳۹۳/۱۰/۰۴ ۱۲:۱۳

سلام
جایی دیدیم که نوعی از حملات هست که هکر تو یکی از پوشه‌های سایت (مثلا 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 CLRF شناخته میشن گویا

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۰۴ ۱۲:۵۴

دسترسی اجرایی را از این پوشه‌ها 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: ۱۳۹۳/۱۰/۰۴ ۱۳:۲۳

در واقع آپلود فایل‌های مخرب نوعی حفره امنیتی در توسعه اپلیکیشن هست که موجب سوء استفاده میتونه 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 (لینک‌های ارجاعی رو مطالعه بفرمایید)

در رابطه با ارجاع کاربران به سایت هدف نوع حمله همون CLRF 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 داده‌های کاربر اعتماد نکنید استوار باشید
2 : تمامی ورودی‌های کاربران را که قرار است مورد مشاهده در عموم و خصوص کاربران در مرورگر باشد تا حد ممکن پاکسازی کنید
مطالعه بیشتر ^

عنوان: **نمایش بلادرنگ اعلامی به تمام کاربران در هنگام درج یک رکورد جدید**
نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۳/۰۷/۱۵ ۱۲:۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, SignalR

در ادامه می‌خواهیم 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 می‌باشند:

دوره "معرفی SignalR و ارتباطات بلادرنگ"

نگاهی به اجزای تعاملی Twitter Bootstrap

ابتدا مدل زیر را در نظر داشته باشید:

```
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 دسته‌بندی یک محصول ساده می‌باشد.
کلاس context نیز به صورت زیر می‌باشد:

```
namespace ShowAlertSignalR.Models
{
    public class ProductDbContext : DbContext
    {
        public ProductDbContext() : base("productSample")
        {
            Database.Log = sql => Debug.Write(sql);
        }
        public DbSet<Product> Products { get; set; }
    }
}
```

همانطور که در ابتدا عنوان شد، می‌خواهیم 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 می‌کنید:

```
[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 متد Index تعریف کرده و سپس مقدار آن را درون این شیء ViewBag انتقال دهیم، این متغییر بیانگر حالتی است که آیا اطلاعات جدیدی برای نمایش وجود دارد یا خیر؟ بنابراین اکشن متد 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 متد Create را به صورتی تغییر می‌دهیم که بعد از درج رکورد موردنظر و هدایت کاربر به صفحه‌ی Index، مقدار این متغییر 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-dismissable'>" +
                "<button type='button' class='close' data-dismiss='alert' aria-
hidden='true'>&times;</button>" +
            "رکورد جدیدی هم اکنون ثبت گردید، برای مشاهده آن صفحه را بروزرسانی کنید" + "</div>");
        };
        $.connection.hub.start().done(function() {
            @{
                if (ViewBag.NotifyUsers)
                {
                    <text>notify.server.sendNotification();</text>
                }
```

```
            }
        });
    </script>
}
```

همانطور که در کدهای فوق مشاهده می‌کنید، بعد از اینکه اتصال با موفقیت برقرار شد (درون متد done) شرط چک کردن متغییر NotifyUsers را بررسی کرده‌ایم. یعنی در این حالت اگر مقدار آن true بود، متد درون هاب را فراخوانی کرده‌ایم. در نهایت پیام به یک div با آی‌دی 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 ؟
احتمالا" 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 GlobalHost.ConnectionManager.GetHubContext استفاده کرد (قسمت ارسال اطلاعات از یک Hub به Hub دیگر در برنامه).

عنوان: **پیاده سازی Template تو در تو در AngularJS و ASP.NET MVC**
نویسنده: میثم نوایی
تاریخ: ۱۳۹۳/۰۷/۱۹ ۱۵:۵۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, AngularJS, formbuilder

در Angular می شود یک سری Template و ساختار از پیش تعریف شده داشت و در هر زمان که نیاز بود مدلی را به آنها پاس داد و نمای HTML مورد نظر را تحویل 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.
ViewModel فرم شما در MVC میتواند چیزی شبیه این باشد

```
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 Angular به اشکال 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 بازگشتی Template را فراخوانی کنید.

```
<script type="text/ng-template" id="ElementTemplate">
    <div ng-if="control.Type == 'JbSection'">
    <h2>{{control.Title}}</h2>
    <ul>
        <li ng-repeat="control in control.Elements" ng-include="'ElementTemplate'"></li>
    </ul>
    </div>
    </script>
```

و یا

```
<script type="text/ng-template" id="element.html">
    {{data.label}}
    <ul>
        <li ng-repeat="element in data.elements" ng-include="'element.html'"></li>
    </ul>
</script>

<ul ng-controller="NestedFormCtrl">
    <li ng-repeat="field in formData" ng-include="'element.html'"></li>
</ul>
```

در اینجا صفحه element.html یک صفحه بیرونی است که Template ما در آن قرار دارد.

عنوان: **یکپارچه سازی سیستم اعتبارسنجی ASP.NET MVC با Kendo UI validator**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۸/۲۷ ۱۱:۵۰
آدرس: www.dotnettips.info
گروه‌ها: JavaScript, MVC, jQuery, Kendo UI

روش پیش فرض اعتبارسنجی برنامه‌های ASP.NET MVC، استفاده از دو افزونه‌ی 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 افزونه‌ی jquery.validate انجام می‌شود و فایل jquery.validate.unobtrusive صرفا یک وفق دهنده و مترجم ویژگی‌های خاص ASP.NET MVC به jquery.validate است.

عدم سازگاری پیش فرض jquery.validate با بعضی از ویجت‌های Kendo UI

در حالت استفاده از Kendo UI، این سیستم هنوز هم کار می‌کند؛ اما با یک مشکل. اگر برای مثال از kendoComboBox استفاده کنید، اعتبارسنجی‌های تعریف شده در برنامه، توسط jquery.validate دیده 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 شده‌است:

```
@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="Sumbit">
                Sumbit
            </button>
        </p>
    </fieldset>
}

@section JavaScript
{
    <script type="text/javascript">
        $(function () {
            $("#Destination").kendoComboBox({
                dataSource: [
                    "loc 1",
                    "loc 2"
                ]
```

```
            });
    </script>
}
```

اگر برنامه را اجرا کنید و بر روی دکمه‌ی submit کلیک نمائید، ویژگی Required عمل نخواهد کرد و عملا در سمت کاربر اعتبارسنجی رخ نمی‌دهد.

```
◢ <div class="editor-field">
  ◢ <span class="k-widget k-combobox k-header k-textbox" style="">
    ◢ <span tabindex="-1" class="k-dropdown-wrap k-state-default" unselectable="on">
        <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>
      ▷ <span tabindex="-1" class="k-select" unselectable="on">…</span>
      </span>
      <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>
    </span>
  ▷ <span class="field-validation-error" data-valmsg-replace="true" data-valmsg-for="Destination">…</span>
  </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 صرفا 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 « اعتبار سنجی ورودی‌های کاربر در Kendo 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، متد 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:

فایل kendo.aspnetmvc.js که در بسته‌ی مخصوص Kendo UI تهیه شده برای ASP.NET MVC موجود است (در پوشه‌ی js آن)، عملکردی مشابه فایل jquery.validate.unobtrusive مایکروسافت دارد. کار آن وفق دادن و ترجمه‌ی اعتبارسنجی unobtrusive به روش 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 remote validation مخصوص Kendo UI را اضافه کرده‌است .

نظرات خوانندگان

نویسنده: امیر
تاریخ: ۱۳۹۳/۰۸/۲۸ ۱۳:۰

سئوالی برای من پیش اومد امکان استفاده از مدل mvc داخل کنترل‌های جاوااسکریپت کندو هست ؟ مثلا اگه combobox انتخاب بشه آیدیش در مدل اصلی انتخاب بشه ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۸/۲۸ ۱۳:۴۶

« استفاده ازExpressionها جهت ایجاد Strongly typed view در ASP.NET MVC »

نویسنده: شاکری
تاریخ: ۱۳۹۳/۱۰/۰۲ ۱۰:۳۷

من برای فعال سازی remote validation از پروژه Moon استفاده کردم اما باز هم عمل نمیکنه!

```
[Moon.Web.Validation.RemoteValidator("IsUserExist", "Admin", HttpMethod = "POST", ErrorMessage = "نام کاربری قبلا 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 (userSerive.isUser(Username)) return Json(true);
            return Json(false);
        }
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۰۳ ۱۵:۲۸

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 isInvalid = 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) {
                                isInvalid = response;
                            }
```

```
                    });
                    return !isInvalid;
                }
            },
            messages: {
                remote: function (input) {
                    return input.data('val-remote');
                }
            }
        });
```

- در اینجا در ابتدا متد remote کتابخانه‌ی jQuery Validator غیرفعال می‌شود. سپس یک rule جدید، به kendoValidator به نام دلخواه remote اضافه شده‌است. چون rule‌های kendoValidator اعمال async را پشتیبانی نمی‌کنند، در درخواست ajax آن async: false تنظیم شده‌است. به این ترتیب سطر پس از ajax، پس از پایان کار عملیات ajax فراخوانی می‌شود و در این حالت kendoValidator 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

مراجعه کنید به مطلب « اعتبار سنجی ورودی‌های کاربر در Kendo UI ». در انتهای مطلب در مورد «اعتبارسنجی سفارشی در DataSource» بحث شده‌است. Kendo UI Grid هم اطلاعات اعتبارسنجی فیلدهای خودش را از data source دریافت می‌کند و ... اصول طراحی اعتبارسنجی آن، هیچ تفاوتی با نکته‌ی عنوان شده ندارد (یک custom rule سفارشی را به نام remote، دقیقا 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 remote validation میشه از مثال 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 isInvalid = 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) {
                                isInvalid = response;
                            }
                        });
                        return !isInvalid;
                    }
                },
                messages: {
                    remote: function(input) {
                        return input.data('val-remote');
                    }
                }
            });
        })(jQuery, kendo);
```

عنوان: ساخت یک Form Generator ساده در MVC
نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۳/۰۹/۱۲ ۱۷:۲۰
آدرس: www.dotnettips.info
گروه‌ها: Entity framework, MVC, Form Generator

در ادامه می‌خواهیم نحوه‌ی ایجاد یک فرم‌ساز ساده را ASP.NET MVC بررسی کنیم.
مدل‌های برنامه ما 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 برنامه ما 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, Radioو... 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>()
    .HasRequired(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:

ویو ایجاد یک فرم:

```
@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>
            <span>عنوان</span>
            <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 ایجاد 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>
            <span>عنوان انگلیسی</span>
            <div>
                @Html.EditorFor(model => model.TitleEn, new { htmlAttributes = new { @class = "form-
```

```
control" } })
                @Html.ValidationMessageFor(model => model.TitleEn, "", new { @class = "text-danger" })
            </div>
        </div>

        <div>
            <span>عنوان فارسی</span>
            <div>
                @Html.EditorFor(model => model.TitleFa, new { htmlAttributes = new { @class = "form-
control" } })
                @Html.ValidationMessageFor(model => model.TitleFa, "", new { @class = "text-danger" })
            </div>
        </div>

        <div>
            <span>نوع فیلد</span>
            <div>
                @Html.EnumDropDownListFor(model => model.FieldType, htmlAttributes: new { @class =
"form-control" })
                @Html.ValidationMessageFor(model => model.FieldType, "", new { @class = "text-danger"
})
            </div>
        </div>

        <div>
            <span>فرم</span>
            <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:

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>
        <div>
            @Html.ActionLink("بازگشت", "Index")
        </div>
    </div>
</div>
```

همانطور که در کدهای فوق مشخص است از اکشن متدی که در ادامه مشاهده خواهید کرد لیستی از فیلدهای مربوط به یک فرم را برای کاربر به صورت رندر شده نمایش داده‌ایم. در اینجا باید براساس فیلد FieldType، نوع فیلد را تشخیص دهیم و المنت متناسب با آن را برای کاربر نهایی رندر کنیم. برای اینکار توسط یک حلقه for در بین تمام فیلدها پیمایش می‌کنیم:

```
for (int i = 0; i < Model.Count(); i++)
{
     // code
}
```

سپس در داخل حلقه یک شرط را برای بررسی نوع فیلد قرار داده‌ایم:

```
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 می‌شود:

```
<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].FieldTyp
```

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 نوشت:

```
[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.SaveAllChanges();
                }
            }
            return View(values);
        }
```

سورس مثال جاری را نیز می‌توانید از اینجا دریافت کنید.

نظرات خوانندگان

نویسنده: مرتضی اسدی
تاریخ: ۱۳۹۳/۰۹/۱۵ ۹:۳۳

با تشکر از مطلب مفید شما
برای اعتبار سنجی فیلدها ایده خاصی دارید؟

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۳/۰۹/۱۵ ۲۲:۶

به دو روش می‌تونید اینکار رو انجام بدید: 1- از یک جدول دیگر برای اعمال اعتبارسنجی استفاده کنید که کاربر خودش بتونه rule اعمال کنه، رابطه این جدول با جدول فیلد هم به صورت یک به چند هست یعنی یک فیلد می‌تونه چند تا validation rule داشته باشه:

```
public class FieldValidation
{
        public int Id { get; set; }
        public string Rule { get; set; }
        public virtual Field Field { get; set; }
}
```

روش دوم:
می‌تونید از یک فیلد اضافی تحت عنوان "متن خطا" 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="عنوان را وارد نمائید" id="[@i].TitleEn" value="" />
                                        <span class="field-validation-valid text-danger" data-valmsg-
for="[@i].TitleEn" data-valmsg-replace="true"></span>
                                    </div>
```

نویسنده: حسین صفدری
تاریخ: ۱۳۹۳/۰۹/۳۰ ۱۷:۳۷

با سلام
با تشکر از مقاله شما
در قسمت مشاهده گزارشات که داده‌ها را می‌خواهیم به صورت گرید kendo نمایش دهیم باید به چه صورت عمل کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۰۱ ۰:۴۹

- 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 (صرفا 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 شده‌است 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 منحصربفرد 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 آن‌را 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 خواهید 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 صفدری
تاریخ: ۱۳۹۳/۱۰/۰۱ ۹:۴۹

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. تغییر DataTable به Json و پاس دادن JSON به Kendo.

نویسنده: نازنین حسینی
تاریخ: ۱۳۹۳/۱۰/۰۱ ۱۴:۱۰

ضمن عرض خسته نباشید خدمت شما
من مدلی مشابه مدل شما دارم با این تفاوت که من به HttpPost کالکشنی از مدل ام را می‌دهم که در مدلم هم یک PropertyCollection دارم که از همین تکنیک hidden استفاده کردم اما باز هم PropertyCollection من دارای Count=0 می‌باشد...
من در واقع کالکشنی از کالکشن دارم...
راه دیگه ای به غیر از hidden برای bind می‌تونم استفاده کنم؟

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۳/۱۰/۰۱ ۱۵:۲۰

View Source 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: ۱۳۹۳/۱۰/۰۳ ۱۱:۲۳

خیلی ممنون.
ViewSource را بررسی کردم پروپرتی name مورد نظر درست ست نمیشد.علتش 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 ست کردم.

عنوان: **آشنایی با WebDav و نحوه استفاده از آن**
نویسنده: رضا بازرگان
تاریخ: ۱۳۹۳/۰۹/۱۳ ۱۸:۴۵
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, MVC, MS Office, webdav, cloud

WebDAV استانداردی است بر روی پروتکل HTTP که Requestها و Responseهای مدیریت یک فایل را بر روی سرویس دهنده وب، تشریح می‌کند.

برای درک چرایی وجود این استاندارد بهتر است ذهن خود را معطوف به نحوه‌ی عملکرد سیستم فایل در OS کنیم که شامل APIهای 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 اسناد Office بر روی وب را ندارید، نیاز است تا اجازه دهید نرم افزارهای Office مستقیما 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 عملا 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، متدی باشد که بتواند به درخواست رسیده، به درستی پاسخ دهد.
این یعنی باید سمت سرور متدی با قابلیت ارسال پاسخ‌های درست و در سمت کلاینت نرم افزاری با قابلیت ارسال درخواست‌های مناسب وجود داشته باشد.
WebDAV استاندارد تشریح محتوای 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 متدی با قابلیت پشتیبانی از درخواست‌های WebDAV پیاده سازی نمایید و البته متاسفانه کتابخانه‌های سورس باز چندانی برای WebDAV در سرور دات نت وجود ندارد. من ماژولی 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 Section 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 ماژول اجرا شوند. در اینجا یعنی درخواست‌هایی که آدرس آن‌ها شامل "/xdav/" باشد، توسط این ماژول Handle می‌شوند. عبارت بعد از مقدار Name در URL هم طبیعتا نام فایل مورد نظر شماست.

FileLocation آدرس پوشه ای است که فایل‌ها در آن ذخیره و یا بازخوانی می‌شوند. اگر FileType با مقدار Local تنظیم شود،

یعنی باید یک پوشه به نام خاصیت URL که در اینجا xdav است در پوشه‌ی اصلی وب شما وجود داشته باشد و اگر با Server مقدار دهی شود URL باید یک آدرس فیزیکی بر روی سرور داشته باشد . مثل "c:\webdav"URL=

ما در این مثال مقادیر را به صورت پیش‌فرض نگه می‌داریم. یعنی باید در پوشه‌ی وب، یک Folder با نام xdav ایجاد کنیم.

در ادامه چند فایل word را برای تست در این پوشه کپی می‌کنم.

می خواهیم در صفحه Index، لیستی از فایل‌های 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 می‌شود 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
```

Webaddress آدرس وبسایت و filename نام فایل مورد نظرمان است. عبارت /xdav/ هم که نشان می‌دهد درخواست هایی که این آدرس را دارند باید توسط ماژول xDav پردازش شوند.

کلاسی با نام DavFile در پوشه‌ی Model ایجاد می‌کنم:

```
public class DavFile
    {
        public string Name { get; set; }
```

```
        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 پوشه‌ی webDav قرار دارند تهیه می‌کنیم و به View ارسال می‌کنیم. View را هم مثل زیر بازنویسی می‌کنیم.

```
@model IEnumerable<WebDavServer.Models.DavFile>
<h1>
    File List
</h1>

<ul>
    @foreach (var item in Model)
    {
        <li> <a href="@Html.Raw(item.Href(ViewContext.HttpContext.Request.Url.Authority))">
@Html.Raw(item.Name) </a></li>
    }
</ul>
```

قرار است به ازای هر فایل، لینکی نمایش داده شود که با کلیک بر روی آن، آدرس فایل به word ارسال می‌شود. بعد از ثبت تغییرات، word محتوا را به همان آدرس ارسال می‌کند و ماژول xDav محتوا را در فایل فیزیکی سرور ذخیره خواهد کرد.

برنامه را اجرا کنید و بر روی فایل‌ها کلیک نمایید. اگر نرم افزار Office روی کامپیوترتان باز باشد با کلیک بر روی هر کدام از فایل‌ها، فایل word باز شده و می‌توانید محتوا را تغییر داده و ذخیره نمایید.

نرم افزار کلاینت (word) درخواست هایی با verbهای مشخص که در استاندارد WebDav ذکر شده به آدرس مورد نظر می‌فرستد. سرور WebDav درخواست را بر اساس Verb آن Request پردازش کرده و Response استاندارد را ایجاد میکند.

نرم افزار word پس از دریافت یک URL، به جای فرمت فیزیکی فایل، درخواست هایی را با تایپ‌های Option, Head, lock, get, post و unlock ارسال می‌کند. محتوای درخواست و پاسخ هر کدام از تایپ‌ها در استاندارد webDav تعریف شده و ماژول xDav آن 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 نمی‌کند و وقتی روی لینک‌های کلیک می‌کنم هیچ اتفاقی نمی‌افتد.

نویسنده: رضا بازرگان
تاریخ: ۱۳۹۳/۰۹/۱۵ ۱۳:۴۶

با سلام.
مطمئن شوید که Office رو سیستم شما نصب است. من Packageها رو از پروژه نمونه حذف کردم . لطفا مجددا آن را نصب کنید.
فرقی در استفاده از مرورگر هم نیست. می‌توانید از پروژه ای که روی Git گذاشتم هم استفاده کنید که کاملتر است.

عنوان: **قابلیت Templated Razor Delegate**
نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۳/۰۹/۲۳ ۱۰:۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, Razor

Razor دارای قابلیتی با نام Templated Razor Delegates است. همانطور که از نام آن مشخص است، یعنی Razor Template هایی که Delegate هستند. در ادامه این قابلیت را با ذکر چند مثال توضیح خواهیم داد. **مثال اول:**
می‌خواهیم تعدادی تگ li را در خروجی رندر کنیم، این کار را می‌توانیم با استفاده از Razor helpers نیز به این صورت انجام دهیم:

```
@helper ListItem(string content) {
 <li>@content</li>
}
<ul>
 @foreach(var item in Model) {
 @ListItem(item)
 }
</ul>
```

همین کار را می‌توانیم توسط Templated Razor Delegate به صورت زیر نیز انجام دهیم:

```
@{
 Func<dynamic, HelperResult> ListItem = @<li>@item</li>;
}
<ul>
 @foreach(var item in Model) {
 @ListItem(item)
 }
</ul>
```

برای اینکار از نوع Func استفاده خواهیم کرد. این Delegate یک پارامتر را می‌پذیرد. این پارامتر می‌تواند از هر نوعی باشد. در اینجا از نوع dynamic استفاده کرده‌ایم. خروجی این Delegate نیز یک HelperResult است. همانطور که مشاهده می‌کنید آن را برابر با الگویی که قرار است رندر شود تعیین کرده‌ایم. 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 میکنند. برای حالت‌هایی مانند کار با آرایه‌ها و یا Enumerations بهتر است از روش دوم استفاده کنید؛ از این جهت که نیاز به کد کمتری دارد و نگهداری آن خیلی از روش اول ساده‌تر 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 می‌کنیم:

```
<header>
    @if (IsSectionDefined("Header"))
    {
        @RenderSection("Header")
    }
    else
    {
        <div>Default Content for Header Section</div>
    }
</header>
```

روش فوق به درستی کار خواهد کرد اما می‌توان آن را با یک خط کد، درون ویو نیز نوشت. در واقع می‌توانیم با استفاده از Templated Razor Delegate یک متد الحاقی برای کلاس ViewPage بنویسیم؛ به طوریکه یک محتوای پیش‌فرض را برای حالتی که

section 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 متد الحاقی 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 Markup آن خواهید 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> را دارد:

```
@model IEnumerable<string>
@{
    ViewBag.Title = "Test";
}
```

و اکشن متد ما 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 یا فرد بودن ردیف‌ها انجام دهیم اما کد زیادی را باید درون ویو بنویسیم. اینکار را می‌توانیم درون یک متد الحاقی نیز انجام دهیم. بنابراین یک متد الحاقی 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> seperatorTemplate = 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 آنجائیکه 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 AlternatingItemtemplate، ItemTemplate و SeperatorItemTemplate تعریف شده‌اند و از نوع Delegate با پارامتر ورودی T و خروجی HelperResult هستند. در داخل متدمان یک HelperResult را برمیگردانیم. 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 برابر با 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 ItemTemplate و AlternatingItemtemplate 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:

```
<table>
    <tr>
        <td>Id</td>
        <td>Name</td>
    </tr>
    @Html.Repeater(Model, @<tr><td>@item.Id</td><td>@item.Name</td></tr>)
</table>
```

عنوان: **اعمال تزریق وابستگی‌ها به مثال رسمی ASP.NET Identity**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۲۹ ۱۳:۳۵
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, Entity framework, MVC, Dependency Injection, ASP.NET Identity

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 پیاده سازی الگوی واحد کار در EF Code First آشنا باشید.
- به علاوه فرض بر این است که یک پروژه‌ی خالی ASP.NET MVC 5 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 پیاده سازی 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.

ساختار پروژه‌ی AspNetIdentityDependencyInjectionSample.DomainClasses

مثال Microsoft.AspNet.Identity.Samples بر مبنای primary key از نوع string است. برای نمونه کلاس کاربران آن‌را به نام ApplicationUser در فایل Models\IdentityModels.cs می‌توانید مشاهده کنید. در مطلب جاری، این نوع پیش فرض، 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 نحوه‌ی تغییر primary key از نوع string را به نوع int، مشاهده می‌کنید. این تغییر نیاز به اعمال به کلاس‌های کاربران و همچنین نقش‌های آن‌ها نیز دارد. به همین جهت صرفا 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 شده‌است 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 آن‌را به پروژه‌ی DomainClasses منتقل کرده‌ایم، مجددا مراجعه کنید، کلاس DbContext مخصوص ASP.NET Identity نیز در آن تعریف شده‌است:

```
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. IdentityDbContext، در اصل از DbContext مشتق شده‌است.
اینترفیس IUnitOfWork برنامه، در پروژه‌ی DataLayer چنین شکلی را دارد که نمونه‌ای از آن‌را 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; }
```

تعریف آن باید جهت اعمال کلاس‌های سفارشی سازی شده‌ی کاربران و نقش‌های آن‌ها برای استفاده از primary key از نوع int 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 متداول 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 ApplicationDbInitializer : DropCreateDatabaseIfModelChanges<ApplicationDbContext>
```

نیازی به این کلاس به این شکل نیست. آن‌را حذف کنید و در پروژه‌ی DataLayer، کلاس جدید ذیل را اضافه نمائید:

```
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.

ساختار 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 عملا معکوس سازی واقعی وابستگی‌ها رخ نداده‌است. بنابراین نیاز است این مسایل را اصلاح کنیم.

الف) انتقال کلاس ApplicationUserManager به لایه سرویس برنامه

کلاس ApplicationUserManager فایل App_Start\IdentityConfig.c 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;

    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 هربار وهله 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 IApplicationUserManager 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 UserManager رسید. کنترلری که با IApplicationUserManager بجای ApplicationUserManager کار می‌کند، قابلیت تعویض پیاده سازی آن‌را جهت آزمون‌های واحد خواهد یافت.

- در کلاس اصلی ApplicationDbInitializer 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 مستقیما 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 IUserStore، IDataProtectionProvider و IIdentityMessageService جزو تعاریف اصلی ASP.NET Identity بوده و نیازی به تعریف مجدد آن‌ها نیست. فقط کلاس‌های EmailService و SmsService فایل App_Start\IdentityConfig.c 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، وهله سازی مستقیم است که مطابق کدهای فوق به حالت تزریق وابستگی‌ها بهبود یافته‌است.

ب) انتقال کلاس ApplicationSignInManager به لایه سرویس برنامه

کلاس ApplicationSignInManager فایل App_Start\IdentityConfig.c را نیز به لایه سرویس منتقل می‌کنیم.

```
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 دقیقا 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 فایل App_Start\IdentityConfig.c 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.

ساختار 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<IApplicationSignInManager>()
                      .HybridHttpOrThreadLocalScoped()
                      .Use<ApplicationSignInManager>();

                ioc.For<IApplicationUserManager>()
                      .HybridHttpOrThreadLocalScoped()
                      .Use<ApplicationUserManager>();

                ioc.For<IApplicationRoleManager>()
                      .HybridHttpOrThreadLocalScoped()
                      .Use<ApplicationRoleManager>();

                ioc.For<IIdentityMessageService>().Use<SmsService>();
                ioc.For<IIdentityMessageService>().Use<EmailService>();
                ioc.For<ICustomRoleStore>()
                      .HybridHttpOrThreadLocalScoped()
                      .Use<CustomRoleStore>();

                ioc.For<ICustomUserStore>()
                      .HybridHttpOrThreadLocalScoped()
                      .Use<CustomUserStore>();

                //config.For<IDataProtectionProvider>().Use(()=> app.GetDataProtectionProvider()); //
In Startup class

                ioc.For<ICategoryService>().Use<EfCategoryService>();
                ioc.For<IProductService>().Use<EfProductService>();
            });
        }
    }
}
```

در اینجا نحوه‌ی اتصال اینترفیس‌های برنامه را به کلاس‌ها و یا نمونه‌هایی که آن‌ها را می‌توانند پیاده سازی کنند، مشاهده می‌کنید.

برای مثال IUnitOfWork به ApplicationDbContext مرتبط شده‌است 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 وهله سازی کند، استفاده می‌شوند و ذکر آن‌ها الزامی است.

در تعاریف فوق یک مورد را به فایل Startup.cs موکول کرده‌ایم. برای مشخص سازی نمونه‌ی پیاده سازی کننده‌ی IDataProtectionProvider نیاز است به IAppBuilder کلاس Startup برنامه دسترسی داشت. این کلاس آغازین Owin اکنون به نحو ذیل بازنویسی شده‌است و در آن، تنظیمات IDataProtectionProvider را به همراه وهله سازی CreatePerOwinContext 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 Startup.Auth.cs پوشه‌ی App_Start 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 متد setDbInitializer، نحوه‌ی استفاده و تعریف فایل Configuration لایه Data را ملاحظه می‌کنید؛ به همراه متد آغاز بانک اطلاعاتی و اعمال تغییرات لازم به آن در ابتدای کار برنامه. همچنین ControllerFactory برنامه نیز به StructureMapControllerFactory 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 int

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 کنترلرها جهت اعمال تزریق وابستگی‌ها ساده‌است. در ادامه نحوه‌ی تغییر امضای سازنده‌های این کنترلرها را 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 متد نیز در اینجا وجود دارند که از string id استفاده می‌کنند. این‌ها را باید به int? Id تغییر داد تا با نوع primary key جدید مورد استفاده تطابق پیدا کنند.

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید:

AspNetIdentityDependencyInjectionSample

نظرات خوانندگان

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۳/۱۰/۰۵ ۱۶:۱۱

ممنون از مطلب شما؛
برای تغییر نام جداول تشکیل شده، درون متد OnModelCreating کد زیر را نوشتم اما با بروز رسانی دیتابیس تغییری در اسامی 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: ۱۳۹۳/۱۰/۰۵ ۱۸:۵۳

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: ۱۳۹۳/۱۰/۰۸ ۱۹:۲۱

با سلام
ممنون از مطلب مفید شما. واقعا 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
        }
```

مخصوصا بخش Dispose .

چون قبلا که از الگوی واحد کار استفاده می‌شد این بخش وجود نداشت !
آیا وجود این کدها که در بالا اومده الزامی است؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۰۸ ۱۹:۲۴

خیر. متد Dispose را حذف کنید (در کلاس پایه هست). وجود سازنده هم صرفا جهت ساده سازی تعریف و انتخاب رشته اتصالی تعریف شده در web.config است.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۱۱ ۱۳:۲۷

نکته‌ای مهم در مورد مدیریت استراکچرمپ در این مثال

اگر از ObjectFactory **Sm** مطلب فوق استفاده می‌کنید، Container آن با ObjectFactory یکی نیست یا به عبارتی ObjectFactory اطلاعی در مورد تنظیمات کلاس سفارشی ObjectFactory **Sm** ندارد. بنابراین دیگر نباید از ObjectFactory قدیمی استفاده کنید.
در این حالت هرجایی ObjectFactory قدیمی را داشتید، با SmObjectFactory.Container تعویض می‌شود.

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۳/۱۰/۱۴ ۱۸:۴۶

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: ۱۳۹۳/۱۰/۱۴ ۱۹:۱۷

User.Identity را در مثال فوق جستجو کنید:

```
User.Identity.GetUserName()
User.Identity.GetUserId()
```

نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۳/۱۰/۱۴ ۱۹:۲۱

برای مشخص کردن نمونه پیاده‌سازی کننده IDataProtectionProvider در یک برنامه کنسول نیز باید از فایل Startup استفاده کرد؟ بیشتر هدفم Seed کردن دیتابیس است (مثلاً ایمپورت تعداد زیادی کاربر از طریق یک فایل و...). اینکار رو در متد SeedDatabase هم انجام دادم ولی هر بار استثنای UserId not found رو در:

```
result = this.SetLockoutEnabled(user.Id, false);
```

صادر میکنه، 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: ۱۳۹۳/۱۰/۱۵ ۱:۱۵

در قسمت تزریق وابستگی کد زیر وجود داره

```
ioc.For<IIdentityMessageService>().Use<SmsService>();
ioc.For<IIdentityMessageService>().Use<EmailService>();
```

در صورت استفاده از اینترفیس مربوطه کدوم سرویس نمونه سازی میشه؟ 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 instanceها 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 متد SeedDatabase استفاده کردم، مشکل این بود که در متد 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<IIdentityMessageService>().Use<SmsService>().Named("smsService");
                ioc.For<IIdentityMessageService>().Use<EmailService>().Named("emailService");
                ioc.For<IApplicationUserManager>().HybridHttpOrThreadLocalScoped()
                    .Use<ApplicationUserManager>()
                    .Ctor<IIdentityMessageService>("smsService").Is<SmsService>()
                    .Ctor<IIdentityMessageService>("emailService").Is<EmailService>()
                    .Setter<IIdentityMessageService>(userManager =>
userManager.SmsService).Is<SmsService>()
                    .Setter<IIdentityMessageService>(userManager =>
userManager.EmailService).Is<EmailService>();
```

و این کد سازنده UserManager

```
public ApplicationUserManager(
            IApplicationUserStore store,
            IApplicationRoleManager roleManager,
            IDataProtectionProvider dataProtectionProvider,
            IIdentityMessageService smsService,
            IIdentityMessageService 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 فراخوانی شده در بار اول تزریق وابستگی درست انجام شده اما در اخرین بار هم سرویس ایمیل به ایمیل و پیامک هر تزریق شده.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۱۶ ۲:۴۵

جهت مدیریت تک وهله‌ای ApplicationUserManger این تغییرات را اعمال کنید؛ یا از این فایل نهایی استفاده کنید.

نویسنده: نیکناز
تاریخ: ۱۳۹۳/۱۰/۲۳ ۱۲:۳۴

سلام؛ در روش برنامه نویسی لایه ای به این صورتی که شما آموزش دادید امکان دیدن جدول‌های بانک در بخش Server Explorer نیست؟ منظورم قبل از اینزرت اطلاعات در آنهاست؟ چون می‌خوام یکسری از جداول رو دستی مستقیم در بانک پر کنم.
و اینکه بعد از اینکه dbContext را در پوشه App_Data حذف کردم چه جوری اتصال کاملشو از بین ببرم چون برنامه ام ارور پیدا نکردن dbcontext را می‌ده و چه جوری می‌تونم دوباره ReCreate کنمش.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۲۳ ۱۳:۲

- در سازنده‌ی کلاس ApplicationDbContext ، به connectionString1 اشاره شده که تعریف آن در فایل web.config برنامه موجود است. به عبارتی در این مثال چنین رشته‌ی اتصالی تعریف شده‌است:

```
<add name="connectionString1"
connectionString="Data Source=(local);Initial Catalog=TestDbIdentity;Integrated Security = true"
providerName="System.Data.SqlClient" />
```

بنابراین بانک اطلاعاتی پیش فرض آن TestDbIdentity نام دارد (جهت اتصال به آن، برای مشاهده جداول و یا تغییر و ثبت اطلاعات). این رشته اتصالی هم مخصوص SQL Server تنظیم شده‌است که می‌توانید توسط management studio و یا سایر ابزارهای مشابه، همانند قبل به آن متصل شوید.
- «در پوشه App_Data حذف کردم» ... از مثال نهایی کامل شده استفاده کنید (^) و نیازی به تکرار این مراحل نیست تا خطای یافت نشدن dbcontext را دریافت نکنید.
- برای ReCreate فقط کافی هست که بانک اطلاعاتی TestDbIdentity را drop کنید. بعد برنامه را مجددا از نو اجرا کنید. چون مراحل migrations آن به حالت خودکار تنظیم شده‌است، بانک اطلاعاتی را به صورت خودکار ایجاد می‌کند یا تغییرات کلاس‌های دومین برنامه را به صورت خودکار به بانک اطلاعاتی اعمال خواهد کرد.
- برای آشنایی بیشتر با مباحث تعریف رشته اتصالی EF Code First و مباحث Migrations آن، سری EF Code First را در سایت یکبار مطالعه کنید.

نویسنده: اقبال
تاریخ: ۱۳۹۳/۱۰/۲۷ ۱۶:۱۰

ممنون از مطالب شما اگر بخواهیم از IApplicationRoleManager و IApplicationUserManager یا به طور کلی از اینترفیس‌ها در AuthorizeAttribute سفارشی (یا فیلتر هایی) که ایجاد کرده ایم استفاده کنیم ، تنظیمات strucutremap به چه صورت خواهد بود.

```
public class CustomAuthorizeAttribute : AuthorizeAttribute
    {


        public IApplicationRoleManager _roleManager { get; set; }

        public IApplicationUserManager _userManager { get; set; }


        public CustomAuthorizeAttribute(IApplicationUserManager userManager,
                                        IApplicationRoleManager roleManager)
        {
            _userManager = userManager;
            _roleManager = roleManager;
        }
        protected override bool AuthorizeCore(HttpContextBase httpContext)
        {
            // Put your custom logic here, returning true for success and false for failure,
            // or return base.AuthorizeCore(httpContext) to defer to the base implementation
            IPrincipal user = httpContext.User;
            IIdentity identity = user.Identity;

            if (!identity.IsAuthenticated)
            {
                return false;
            }

            bool isAuthorized = true;
            // TODO: perform custom authorization against the App
            var qry = _userManager.Users.Where(u => u.UserName == identity.Name).ToList();

            return isAuthorized;
        }
    }
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۲۷ ۱۷:۴۱

« تزریق وابستگی‌ها در فیلترهای ASP.NET MVC »

نویسنده: م سلیمانی
تاریخ: ۱۳۹۳/۱۱/۰۷ ۱۲:۳۳

با سلام؛ پروژه این مثال و دانلود کردم و سپس اجراش کردم - مشکلی نداره ولی بعد چند دقیقه ناگهان با این پیغام مواجه میشم.

لطفا راهنمایی بفرمائید. با تشکر.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۱/۰۷ ۱۳:۲۳

- لطفا خطاها را تصویری ارسال نکنید. متن موجود در تصاویر، قابلیت جستجو و پیگیری ندارند.
- متن آن واضح است و به عمد تنظیم شده‌است. عنوان کرده‌است که آدرس خاصی را یافت نمی‌کند و این آدرس‌ها قابلیت تزریق وابستگی ندارند. آدرس آن (arterySignalR/ping) مرتبط است به browser link که به این صورت می‌توانید آن‌را غیرفعال کنید:

```
public static void RegisterRoutes(RouteCollection routes)
{
    routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
    routes.IgnoreRoute("{*browserlink}", new { browserlink = @".*/arterySignalR/ping" });
    //...
}
```

و یا

```
<appSettings>
    <add key="vs:EnableBrowserLink" value="false" />
</appSettings>
```

نویسنده: س محمدرضا برنتی
تاریخ: ۱۳۹۴/۰۱/۱۶ ۱۶:۳۸

تنظیمات جهت اعمال تزریق وابستگی‌ها در WEB API چگونه خواهد بود ؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۴/۰۱/۱۶ ۱۶:۵۷

تنظیمات اولیه‌ی IoC Container در این حالت، با نمونه‌ی موجود یکی است. فقط Web API نیاز به معرفی SmObjectFactory.Container را به GlobalConfiguration مخصوص خودش، به صورت جداگانه دارد. اطلاعات بیشتر

نویسنده: سروش شیرالی
تاریخ: ۱۳۹۴/۰۱/۲۸ ۱۸:۳۵

با سلام؛ خیلی معماری جالبی بود. اما یه سوالی ذهن مرا مشغول کرده . البته من ساختار structureMap را درست نمی‌دانم ولی اکثر IOC Containerها نمونه singlethon را از اینترفیس‌ها برمی گردانند . که اگر اینگونه باشد به معنی وجود یک Unit OF Work و DbContext بوده در کل appllication بوده که این برای یک وب اپلیکیشن به منزله یک فاجعه است. اگر ممکن است مقداری راهنمایی بفرمایید. سپاسگزارم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۴/۰۱/۲۸ ۱۹:۶

طول عمر پیش فرض اشیاء ایجاد شده‌ی توسط استراکچرمپ از نوع transient است. به این معنا که هر درخواستی از آن شیء، سبب ایجاد یک وهله‌ی جدید از آن خواهد شد. اطلاعات بیشتر

نویسنده: سروش شیرالی
تاریخ: ۱۳۹۴/۰۱/۳۰ ۱۷:۵۸

فرض کنید ما از این پروژه استفاده کنیم . اما از الگوی repository و unit of work بر روی آنها نیز استفاده کنیم. به این ترتیب IOC از یک طرف applicationDbContext را به کلاس store در identity پاس داده و از طرف دیگر به کلاس unit of work آیا به نظر شما مشکلی پیش نخواهد آمد.
با توجه به آنکه به فرموده شما IOC در هر HttpRequest فقط یک آبجکت از کلاس applicationDbContext خواهد ساخت اگر store و unit of work هر دو در همان HttpRequest اقدام به اعمال savechange کنند چه؟ اصلا چینین چیزی ممکن است؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۴/۰۱/۳۰ ۱۸:۳۲

context و uow یکی هستند. قسمت‌های 11 و 12 سری EF Code first سایت را نیاز است یکبار مطالعه کنید؛ به همراه تمام نظرات آن‌ها.

عنوان: ثبت جزئیات استثناهای Entity framework توسط ELMAH
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۰۸ ۱۱:۲۰
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, Entity framework, MVC, ELMAH

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.

کلاس EfExceptionsInterceptor

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 شده‌است.

```
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 آن‌را جهت استفاده از ELMAH به نحو ذیل ارائه داد:

```
using System;
using System.Data.Common;
using System.Data.Entity.Infrastructure.Interception;
using Elmah;

namespace ElmahEFLogger.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 رخ داده را به همراه دارد. می‌توان این اطلاعات را پس از اندکی نظم بخشیدن، به متد Raise مربوط به ELMAH ارسال کرد تا جزئیات بیشتری از استثنای رخ داده شده، در لاگ‌های 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 ابتدای بحث شروع کرد و وهله‌ای از ElmahEfExceptionsLogger را به سازنده‌ی آن تزریق نمود (یکی از چندین روش ممکن). سپس برای

استفاده از آن کافی است به ابتدای متد Application_Start فایل Global.asax.cs مراجعه و در ادامه سطر ذیل را اضافه نمود:

```
DbInteception.Add(new ElmahEfInterceptor());
```

پس از آن جزئیات کلیه استثناهای EF در لاگ‌های نهایی ELMAH به نحو ذیل ظاهر خواهند شد:

کدهای کامل این پروژه را از اینجا می‌توانید دریافت کنید:

ElmahEFLogger

نظرات خوانندگان

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۳/۱۰/۰۸ ۲۰:۴۷

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 بشه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۰۸ ۲۱:۲۵

خیر. کار این Interceptor صرفا لاگ کردن سراسری و نامرئی ریز جزئیات استثناهای EF در کل برنامه است. استثناهای رخ داده، در ادامه از فیلتر ELMAH اصلی تنظیم شده رد خواهند شد و یکبار هم در آنجا لاگ می‌شوند. این مورد چون منحصر به EF نیست، بنابراین ضرورتی به حذف آن هم نیست.

عنوان: **نمایش بلادرنگ اعلامی به تمام کاربران در هنگام درج یک رکورد جدید به صورت notification**
نویسنده: سیروان عفیفی
تاریخ: ۱۳۹۳/۱۰/۱۲ ۱۳:۴۵
آدرس: 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 بلافاصله 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 - info");
toastr.success("نمایش یک پیام - success");
toastr.error("نمایش یک پیام - error");
toastr.warning("نمایش یک پیام - warning");
```

دستورات فوق خروجی‌های زیر را نمایش می‌دهد:

برای پیام‌های فوق نیز می‌توانید عنوانی را انتخاب کنید:

```
toastr.success("نمایش یک پیام - success", "عنوان");
```

اگر به فایل 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>&times;</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 محض 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، بلافاصله 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>
                }
```

```
            }
});
```

همانطور که مشاهده می‌کنید از onClick برای toastr استفاده کرده‌ایم. با این callback 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 جاوااسکریپتی 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: ۱۳۹۳/۱۰/۱۲ ۱۷:۱۱

نظرتون درباره استفاده از SqlDependency برای اینکار چیه؟ فکر می‌کنید توی یک پروژه بزرگ استفاده از کدامیک بهتر است؟

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۱۰/۱۲ ۲۱:۴۵

SqlDependency 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 API همون ORM استفاده می‌کنید، با عوض کردن پروایدر اون، می‌تونید 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: ۱۳۹۳/۱۰/۱۲ ۲۱:۵۳

اگر بخواهیم در یک ویندوز سرویس متوجه تغییرات بشیم چطور؟ یعنی هیچ واسط دیگه ای وجود نداشته باشه

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۱۰/۱۲ ۲۲:۳۸

SignalR محدود به وب نیست: نگاهی به گزینه‌های مختلف مهیای جهت میزبانی SignalR (در مورد سرور) و کلاینت دات نتی هم می‌تونه داشته باشه: نگاهی به SignalR Clients . حتی کلاینت جاوایی هم می‌تونه داشته باشه: استفاده از SignalR در اندروید

عنوان: **کار با وب سرویس جاوایی تشخیص ایمیل‌های موقتی در دات نت**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۰/۲۷ ۱۲:۴۰
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, MVC, Security, WCF, xml, Validation, Java

یکی از وب سرویس‌های سایت name api ، امکان تشخیص موقتی بودن ایمیل مورد استفاده‌ی جهت ثبت نام در یک سایت را فراهم می‌کند. آدرس WSDL آن نیز در اینجا قرار دارد. اگر مطابق معمول استفاده از سرویس‌های وب در دات نت، بر روی ارجاعات پروژه کلیک راست کرده و گزینه‌ی Add service refrence را انتخاب کنیم و سپس آدرس WSDL یاد شده را به آن معرفی کنیم، بدون مشکل ساختار این وب سرویس دریافت و برای استفاده‌ی از آن به یک چنین کدی خواهیم رسید:

```
var client = new SoapDisposableEmailAddressDetectorClient();

var context = new soapContext
{
    //todo: get your API key here: http://www.nameapi.org/en/register/
    apiKey = "test"
};
var result = client.isDisposable(context, "DaDiDoo@mailinator.com");

if (result.disposable.ToString() == "YES")
{
    Console.WriteLine("YES! It's Disposable!");
}
```

متد isDisposable ارائه شده‌ی توسط این وب سرویس، دو پارامتر context که در آن باید API Key خود را مشخص کرد و همچنین آدرس ایمیل مورد بررسی را دریافت می‌کند. اگر به همین ترتیب این پروژه را اجرا کنید، با خطای Bad request از طرف سرور متوقف خواهید شد:

```
Additional information: The remote server returned an unexpected response: (400) Bad Request.
```

اگر به خروجی این وب سرویس در فیدلر مراجعه کنیم، چنین شکلی را خواهد داشت:

```
<html><head><title>Bad Request</title></head><body><h1>Bad Request</h1><p>No api-key
provided!</p></body></html>
```

عنوان کرده‌است که api-key را، در درخواست وب خود ذکر نکرده‌ایم.

اگر همین وب سرویس را توسط امکانات سایت http://wsdlbrowser.com بررسی کنید، بدون مشکل کار می‌کند. اما تفاوت در کجاست؟

خروجی ارسالی به سرور، توسط سایت http://wsdlbrowser.com به این شکل است:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://disposableemailaddressdetector.email.services.v4_0.soap.server.nameapi.org/">
  <SOAP-ENV:Body>
    <ns1:isDisposable>
      <context>
        <apiKey>test</apiKey>
      </context>
      <emailAddress>sdsdg@site.com</emailAddress>
    </ns1:isDisposable>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

و نمونه‌ی تولید شده‌ی توسط WCF (امکان Add service reference در حقیقت یک WCF Client را ایجاد می‌کند) به صورت زیر می‌باشد:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
     <isDisposable
```

```
xmlns="http://disposableemailaddressdetector.email.services.v4_0.soap.server.nameapi.org/">
          <context xmlns=""><apiKey>test</apiKey></context>
          <emailAddress xmlns="">DaDiDoo@mailinator.com</emailAddress>
      </isDisposable>
  </s:Body>
</s:Envelope>
```

از لحاظ اصول XML، خروجی تولیدی توسط WCF هیچ ایرادی ندارد. از این جهت که نام فضای نام مرتبط با http://schemas.xmlsoap.org/soap/envelope/ را به s تنظیم کرده‌است و سپس با استفاده از این نام، Envelope را تشکیل داده‌است. اما ... این وب سرور جاوایی دقیقا با نام SOAP-ENV کار می‌کند و فضای نام ns1 بعدی آن. کاری هم به اصول XML ندارد که باید بر اساس نام xmlns ذکر شده، کار Parse ورودی دریافتی صورت گیرد و نه بر اساس یک رشته‌ی ثابت از پیش تعیین شده. بنابراین باید راهی را پیدا کنیم تا بتوان این s را تبدیل به SOAP-ENV کرد.

برای این منظور به سه کلاس ذیل خواهیم رسید:

```
public class EndpointBehavior : IEndpointBehavior
{
    public void AddBindingParameters(ServiceEndpoint endpoint, BindingParameterCollection
bindingParameters)
    { }

    public void ApplyDispatchBehavior(ServiceEndpoint endpoint, EndpointDispatcher endpointDispatcher)
    { }

    public void Validate(ServiceEndpoint endpoint)
    { }

    public void ApplyClientBehavior(ServiceEndpoint endpoint, ClientRuntime clientRuntime)
    {
        clientRuntime.MessageInspectors.Add(new ClientMessageInspector());
    }
}

public class ClientMessageInspector : IClientMessageInspector
{
    public void AfterReceiveReply(ref Message reply, object correlationState)
    { }

    public object BeforeSendRequest(ref Message request, System.ServiceModel.IClientChannel channel)
    {
        request = new MyCustomMessage(request);
        return request;
    }
}

/// <summary>
/// To customize WCF envelope and namespace prefix
/// </summary>
public class MyCustomMessage : Message
{
    private readonly Message _message;

    public MyCustomMessage(Message message)
    {
        _message = message;
    }

    public override MessageHeaders Headers
    {
        get { return _message.Headers; }
    }

    public override MessageProperties Properties
    {
        get { return _message.Properties; }
    }

    public override MessageVersion Version
    {
        get { return _message.Version; }
    }

    protected override void OnWriteStartBody(XmlDictionaryWriter writer)
    {
```

```
        writer.WriteStartElement("Body", "http://schemas.xmlsoap.org/soap/envelope/");
    }

    protected override void OnWriteBodyContents(XmlDictionaryWriter writer)
    {
        _message.WriteBodyContents(writer);
    }

    protected override void OnWriteStartEnvelope(XmlDictionaryWriter writer)
    {
        writer.WriteStartElement("SOAP-ENV", "Envelope", "http://schemas.xmlsoap.org/soap/envelope/");
        writer.WriteAttributeString("xmlns", "ns1", null, value:
"http://disposableemailaddressdetector.email.services.v4_0.soap.server.nameapi.org/");
    }
}
```

که پس از تعریف client به نحو ذیل معرفی می‌شوند:

```
var client = new SoapDisposableEmailAddressDetectorClient();
client.Endpoint.Behaviors.Add(new EndpointBehavior());
```

توسط EndpointBehavior سفارشی، می‌توان به متد **Envelope** OnWriteStart دسترسی یافت و سپس s آن‌را با SOAP-ENV درخواستی این وب سرویس جایگزین کرد. اکنون اگر برنامه را اجرا کنید، بدون مشکل کار خواهد کرد و دیگر پیام یافت نشدن API-Key را صادر نمی‌کند.

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید.

عنوان: **فراخوانی متدهای Controllerها در Viewهای ASP.NET MVC**
نویسنده: تایماز قبادی
تاریخ: ۱۳۹۳/۱۱/۰۶ ۱۵:۱۵
آدرس: www.dotnettips.info
گروه‌ها: C#, MVC, Delegate, Func

شاید در ابتدا فراخوانی متدی از یک کنترلر در یک View کار سختی به نظر برسد، ولی در واقع با استفاده از مفاهیم Lambda expressions و Delegateها این کار بسیار راحت خواهد بود.

برای این کار میتوانیم متد مورد نظر را به صورت یک delegate تعریف کرده و به view ارسال کنیم. فرض کنیم متدی داریم برای برگرداندن مجموع 2 عدد به صورت string:

```
public string Sum(int a,int b)
{
    return (a + b).ToString();
}
```

حال برای اینکه بتوانیم این متد را بصورت یک delegate به view ارسال کنیم لازم است تا یک delegate را بصورت public و در خارج از تعریف کلاسها و درون یک namespace مشخصی تعریف کنیم. در اینجا برای راحتی در همان MvcTest.Controllers namespace یک delegate را بصورت زیر (MvcTest نام پروژه است) تعریف می‌کنیم:

```
public delegate string SumOf2Number(int a, int b);
```

حال میتوانیم بصورت زیر این متد را از طریق ViewBag به View ارسال کنیم:

```
SumOf2Number sum2numbers = Sum;
ViewBag.SumFunc3 = sum2numbers;
```

در روش دوم، می‌توانیم متد مورد نظر را بصورت Func به View ارسال کنیم. این کار را میتوانیم به دو صورت انجام دهیم، که هر دو را در تکه کد زیر خواهید دید:

```
ViewBag.SumFunc =(Func<int,int,string>) Sum;//way 1
ViewBag.SumFunc2 = (Func<int, int, string>)((int a, int b) => { return (a + b).ToString(); });//way 2
```

همانطور که متوجه شدید، در روش اول تنها کاری که کردیم متد Sum را از طریق TypeCasting به یک Func تبدیل کردیم و در روش دوم هم یک Lambda expression را بصورت مستقیم به Func تبدیل کرده و استفاده کردیم.

میتوانیم یک Lambda expression را به یک متغیر delegate نیز ربط دهیم؛ به این صورت:

```
SumOf2Number sum2numbers2 = (int a, int b) => { return (a + b).ToString(); };
```

در نهایت کد بخش کنترلر کلاً به اینصورت خواهد بود:

```
namespace MvcTest.Controllers
{
    public  delegate string SumOf2Number(int a, int b);
```

```
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            SumOf2Number sum2numbers = Sum;
            SumOf2Number sum2numbers2 = (int a, int b) => { return (a + b).ToString(); };

            ViewBag.SumFunc =(Func<int,int,string>) Sum;
            ViewBag.SumFunc2 = (Func<int, int, string>)((int a, int b) => { return (a + b).ToString();
});
            ViewBag.SumFunc3 = sum2numbers;
            ViewBag.SumFunc4 = sum2numbers2;
            return View();
        }
        public string Sum(int a,int b)
        {
            return (a + b).ToString();
        }
        public ActionResult About()
        {
            ViewBag.Message = "Your application description page.";

            return View();
        }

        public ActionResult Contact()
        {
            ViewBag.Message = "Your contact page.";

            return View();
        }
    }
}
```

و در Index View خواهیم داشت: (البته اصولاً استفاده از controller namespace در سمت view کار درستی نیست، منتها اینجا فقط یک مثال کاربردی ساده است)

```
@using MvcTest.Controllers;
@{
    ViewBag.Title = "Home Page";
}

<h1>
    @ViewBag.SumFunc(7,8)
</h1>
<h1>
    @ViewBag.SumFunc2(9, 10)
</h1>
<h1>
    @ViewBag.SumFunc3(5, 1)
</h1>
<h1>
    @ViewBag.SumFunc4(2, 3)
</h1>
```

عنوان: **لغو اجرای یک اکشن فیلتر برای یک اکشن خاص در MVC**
نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۳/۱۱/۰۷ ۲۲:۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, ASP.Net MVC, Attributes, ActionFilters

بسیار پیش می‌آید که یک کنترلر را به یک اکشن فیلتر خاص مزین کنیم. در این صورت تمامی اکشن‌های موجود در کنترلر مربوطه مجاب به اجرای اکشن فیلتر مورد نظر می‌شوند. اما بسیار پیش می‌آید که نخواهیم یک اکشن خاص در کنترلر مذکور اکشن فیلتر مورد نظر را اجرا کند.
یک راهکار ساده اما (به نظر شخصی من) غیر منطقی این است که تک تک اکشن هایی را که می‌خواهیم اکشن فیلتر مورد نظر را اجرا کنند، مزین کنیم و اکشن‌هایی که نمی‌خواهیم اکشن فیلتر مورد نظر را اجرا کنند به اکشن فیلتر مورد نظر مزین نمی‌کنیم. اما فرض کنید تعداد اکشن‌های ما زیاد باشند؛ به نظر این روش غیر منطقی و غیر بهینه است.

یکی از مشکلاتی که در یکی از پروژه‌ها حدود سه روز وقت من را گرفت همین کار بود. زمانی که از یک تصویر امنیتی جهت مقابله با ربوت‌های استفاده میکردم به این مشکل برخورد کردم. کنترلر من به یک اکشن فیلتر فشرده سازی محتوا مزین بود. در نتیجه اکشنی که تصویر امنیتی را تولید میکرد نیز اکشن فیلتر فشرده سازی را اجرا می‌کرد و پس از فشرده سازی باعث می‌شد که تصویر امنیتی نشان داده نشود، چون فرمت تصویری آن بهم ریخته بود. یک راهکار را که پس از جستجو به آن رسیدم، جهت استفاده‌ی دوستان مطرح می‌کنم.

برای اینکه از اجرای چنین اکشن فیلتر هایی جلوگیری کنیم نیاز است کمی اکشن فیلتر مورد نظر را دستکاری کنیم.
برای این کار باید مراحل زیر را انجام داد:
1- ابتدا یک Attribute خالی را تعریف می‌کنیم.
2- سپس اکشن فیلتر دلخواهی را تعریف کرده و در زمان اجرا بررسی می‌کنیم اگر متد (اکشن) مورد نظر با Attribute تعریفی در مرحله یک مزین شده بود، در نتیجه اکشن فیلتر را اجرا نمی‌کنیم.
3- هر اکشنی را که نمی‌خواهیم اکشن فیلتر تعریفی مرحله 2 را اجرا کند، آن را به Attribute مرحله یک مزین میکنیم.
به این ترتیب می‌توانیم از اجرای اکشن فیلتر دلخواه روی متدها یا اکشن‌های دلخواه جلوگیری کنیم. در ادامه نحوه‌ی تعریف آنها را در زیر مشاهده میکنید.

1- تعریف یک Attribute دلخواه مثلا با نام DisableCompression

```
[AttributeUsage(AttributeTargets.Class | AttributeTargets.Method, AllowMultiple = false, Inherited =
true)]
public sealed class DisableCompression : Attribute { }
```

2- تعریف اکشن فیلتر دلخواه مثلا با نام CompressionFilter

```
[AttributeUsage(AttributeTargets.Class | AttributeTargets.Method, Inherited = true, AllowMultiple =
false)]
public class CompressionFilter : ActionFilterAttribute
{
    public override void OnActionExecuting(ActionExecutingContext filterContext)
    {
        bool disabled = filterContext.ActionDescriptor.IsDefined(typeof(DisableCompression), true) ||

filterContext.ActionDescriptor.ControllerDescriptor.IsDefined(typeof(DisableCompression), true);
        if (disabled)
            return;

        // کدهای دلخواه اکشن فیلتر مورد نظر
    }
}
```

3- در این مرحله هر اکشنی را که نمی‌خواهیم اکشن فیلتر CompressionFilter را اجرا کند به Attribute با نام

DisableCompression مزین میکنیم.

```
[CompressionFilter]
public abstract class BaseController : Controller
{
}

public class SomeController : BaseController
{
    public ActionResult WantThisActionCompressed()
    {
        // code
    }

    [DisableCompression]
    public ActionResult DontWantThisActionCompressed()
    {
        // code
    }
}
```

با این کار اکشن WantThisActionCompressed اکشن فیلتر CompressionFilter را اجرا می‌کند اما اکشن DontWantThisActionCompressed چون مزین به DisableCompression شده‌است، پس در نتیجه اکشن فیلتر CompressionFilter بر روی آن اجرا نخواهد شد.

عنوان: **کار با اسکنر در برنامه های تحت وب (قسمت اول)**
نویسنده: محمد رعیت پیشه
تاریخ: ۱۳۹۳/۱۱/۱۷ ۰:۱۵
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, MVC, WCF, TWAIN, Cross-Origin Request

در اکثر برنامه‌های سازمانی، مثل برنامه‌های مدیریت آرشیو اسناد، همواره این نیاز جزو خواسته‌های کاربران بوده که بتوانند به صورت مستقیم و از طریق تنها یک کلیک، تصویر مورد نظر را اسکن کرده و به صورت خودکار در برنامه وارد کنند؛ یعنی بدون اینکه نیاز باشد با استفاده از یک برنامه دیگر ابتدا تصویر را اسکن کرده و سپس در فرم وب، فایل اسکن شده را Browse کنند.
این نیاز اساسا به معنی دسترسی به سخت افزار کاربر از طریق مرورگر می‌باشد که به دلایل متعددی امکان پذیر نیست! مشکلات امنیتی ایجاد شده در این راه بسیار جدی است. اما خوشبختانه راههایی برای رسیدن به این هدف وجود دارند:

1- **ActiveX** : که به صورت native فقط در IE پشتیبانی می‌شود (در نسخه‌های جدیدتر IE نیاز به بروز رسانی پلاگین ActiveX controls می‌باشد) و برای استفاده‌ی از آن در مرورگرهای Firefox و Chrome هم باید از پلاگین‌های جانبی روی هر مرورگر استفاده کرد که مثلا برای اجرای بر روی Firefox، باید افزونه‌ی IE Tab را نصب کرد که تنها کارش این است که سایت را از طریق موتور IE در پنجره‌ی فایرفاکس اجرا کند! که البته این مورد مثل این می‌ماند که سایت در IE باز شده باشد که ممکن است زیاد خوشایند نباشد؛ در غیر اینصورت باید پروژه را از اول بر مبنای اجرای بر روی IE طراحی و پیاده سازی کرد. در ضمن از مشکلات اجرای پلاگین نوشته شده توسط برنامه نویس در IE و نسخه‌های مختلف آن هم چشم پوشی می‌کنیم. یکی از مزیت‌های این روش نسبت به بقیه این هست که دانلود و نصب پلاگین مورد نیاز به صورت خودکار و توسط مرورگر انجام می‌شود.

2- **استفاده از یک کامپوننت جانبی** : مثل کامپوننت‌های LEADTOOLS که ابزارهای متنوع و SDKهای بسیار قدرتمندی را برای اینکار و کارهای دیگر، مانند کار با اسکن تصاویر مغزی، پردازش تصویر، بارکد خوان‌ها، مدیریت اسناد و ... فراهم کرده است. قیمت این ابزار بسیار زیاد است و در برخی موارد امکانات فراهم آورده بسیار بیشتر است از خواسته‌ی ما. این ابزار، هم در HTML & Javascript و هم در DotNet قابل استفاده است و مستندات نسبتا خوبی هم در این زمینه ارائه کرده است. همچنین کامپوننت Dynamsoft که باز هم غیر رایگان هست و قیمت بالایی نیز دارد.

اگر روال کار کامپوننت‌های بالا را مورد بررسی قرار دهید (از طریق اجرای Demo ها، اینجا و اینجا) متوجه خواهید شد که هر دو نیاز به نصب یک سرویس یا App سمت کلاینت جهت اجرای دستورات خود دارند. پس می‌شود اینطور نتیجه گرفت که انجام اینکار بدون اینکه چیزی سمت کاربر نصب شود، ممکن نیست و در هر دو، لینک نصب فایل exe سرویس برای دانلود قرار داده شده است. بر این اساس به راه حل سومی خواهیم رسید که خودمان این سرویس را جهت تعامل با اسکنر سمت کاربر طراحی و پیاده سازی نماییم.

اما چطور ممکن است که با اجرای یک فایل exe سمت کاربر (با این فرض که کاربر در یک دامین مطمئن قرار دارد و می‌شود درخواست نصب سرویس را نمود) این امکان را برای کاربر فراهم نمود که با یک کلیک در مرورگر، اسکنر به صورت خودکار اسکن را آغاز کرده و سپس تصویر حاصل را به یکی از کنترلرهای ما در سمت سرور ارسال نماید؟

برای اینکار ما با دو صورت مساله مواجه هستیم؛ اول اینکه چطور تصویر را سمت کاربر اسکن کنیم و دوم اینکه چطور این تصویر را به سرور ارسال نماییم!
برای مساله‌ی اول از کتابخانه (Windows Image Acquisition (WIA استفاده خواهیم نمود که این کتابخانه به ما این امکان را میدهد تا با سخت افزارهایی که از TWAIN پشتیبانی می‌کنند، بتوانیم ارتباط برقرار نماییم.

برای مساله‌ی دوم هم نیاز به پیاده سازی یک WCF Service و اجرای آن (هاست کردن) در سمت کلاینت داریم. در واقع با صدا زدن متدهای این سرویس، از کتابخانه‌ی بالا استفاده کرده و اسکن را انجام می‌دهیم.

ادامه دارد...

عنوان: **کار با اسکنر در برنامه های تحت وب (قسمت دوم و آخر)**
نویسنده: محمد رعیت پیشه
تاریخ: ۱۳۹۳/۱۱/۲۰ ۲۳:۳۵
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, MVC, WCF, TWAIN, Cross-Origin Request, Scanning

در قسمت قبل « کار با اسکنر در برنامه‌های تحت وب (قسمت اول) » دیدی از کاری که قرار است انجام دهیم، رسیدیم. حالا سراغ یک پروژه‌ی عملی و پیاده سازی مطالب مطرح شده می‌رویم.

ابتدا پروژه‌ی WCF را شروع می‌کنیم. ویژوال استودیو را باز کرده و از قسمت New Project > Visual C# > WCF یک پروژه‌ی WCF Service Application جدید را مثلا با نام "WcfServiceScanner" ایجاد نمایید. پس از ایجاد، دو فایل IService1.cs و Service1.scv موجود را به IScannerService و ScannerService تغییر نام دهید. سپس ابتدا محتویات کلاس اینترفیس IScannerService را به صورت زیر تعریف نمایید :

```
[ServiceContract]
    public interface IScannerService
    {
        [OperationContract]
        [WebInvoke(Method = "GET",
           BodyStyle = WebMessageBodyStyle.Wrapped,
           RequestFormat = WebMessageFormat.Json,
           ResponseFormat = WebMessageFormat.Json,
           UriTemplate = "GetScan")]
        string GetScan();
    }
```

در اینجا ما فقط اعلان متدهای مورد نیاز خود را ایجاد کرده‌ایم. علت استفاده از Attribute ایی با نام WebInvoke ، مشخص نمودن نوع خروجی به صورت Json است و همچنین عنوان آدرس مناسبی برای صدا زدن متد. پس از آن کلاس ScannerService را مطابق کدهای زیر تغییر دهید:

```
public class ScannerService : IScannerService
    {
        public string GetScan()
        {
            // TODO Add code here
        }
    }
```

تا اینجا فقط یک WCF Service معمولی ساخته‌ایم .در ادامه به سراغ کلاس WIA برای ارتباط با اسکنر می‌رویم.
بر روی پروژه‌ی خود راست کلیک کرده و Add Reference را انتخاب نموده و سپس در قسمت COM، گزینه‌ی Microsoft Windows Image Acquisition Library v2.0 را به پروژه‌ی خود اضافه نمایید.
با اضافه شدن این ارجاع به پروژه، دسترسی به فضای نام WIA برای ما امکان پذیر می‌شود که ارجاعی از آن را در کلاس ScannerService قرار می‌دهیم.

```
using WIA;
```

اکنون متد GetScan را مطابق زیر اصلاح می‌نماییم:

```
public string GetScan()
        {
            var imgResult = String.Empty;
            var dialog = new CommonDialogClass();
            try
            {
                // نمایش فرم پیشفرض اسکنر
                var image = dialog.ShowAcquireImage(WiaDeviceType.ScannerDeviceType);

                // ذخیره تصویر در یک فایل موقت
                var filename = Path.GetTempFileName();
```

```
            image.SaveFile(filename);
            var img = Image.FromFile(filename);

            // img تبدیل تصویر به Base64 جهت ارسال سمت کاربر و نمایش در تگ
            imgResult = ImageHelper.ImageToBase64(img, ImageFormat.Jpeg);
        }
        catch
        {
            // از آنجاییه که امکان نمایش خطا وجود ندارد در صورت بروز خطا رشته خالی
            // بازگردانده می‌شود که به معنای نبود تصویر می‌باشد
        }

        return imgResult;
    }
```

دقت داشته باشید که کدها را در زمان توسعه بین Try..Catch قرار ندهید چون ممکن‌است در این زمان به خطاهایی برخورد کنید که نیاز باشد در مرورگر آنها را دیده و رفع خطا نمایید.

CommonDialogClass کلاس اصلی در اینجا جهت نمایش فرم کار با اسکنر می‌باشد و متدهای مختلفی را جهت ارتباط با اسکنر در اختیار ما قرار می‌دهد که بسته به نیاز خود می‌توانید از آنها استفاده کنید. برای نمونه در مثال ما نیز متد اصلی که مورد استفاده قرار گرفته، ShowAcquireImage می‌باشد که این متد، فرم پیش فرض دریافت اسکنر را به کاربر نمایش می‌دهد و کاربر از طریق آن می‌تواند قبل از شروع اسکن، یکسری تنظیمات را انجام دهد.

این متد ابتدا به صورت خودکار فرم تعیین دستگاه اسکنر ورودی را نمایش داده :

و سپس فرم پیش فرض اسکنرهای TWAIN را جهت تعیین تنظیمات اسکن نمایش می‌دهد که این امکان نیز در این فرم فراهم است تا دستگاه‌های Feeder یا Flated انتخاب گردند.

خروجی این متد همان عکس اسکن شده است که از نوع WIA.ImageFile می‌باشد و ما پس از دریافتش، ابتدا آن را در یک فایل موقت ذخیره نموده و سپس با استفاده از یک متد کمکی آن را به فرمت Base64 برای درخواست کننده اسکن ارسال می‌نماییم.

کدهای کلاس کمکی ImageHelper:

```
public static string ImageToBase64(Image image, System.Drawing.Imaging.ImageFormat format)
        {
            if (image != null)
            {
                using (MemoryStream ms = new MemoryStream())
                {
                    // Convert Image to byte[]
                    image.Save(ms, format);
                    byte[] imageBytes = ms.ToArray();

                    // Convert byte[] to Base64 String
                    string base64String = Convert.ToBase64String(imageBytes);
                    return base64String;
                }
            }
            return String.Empty;
        }
```

توجه داشته باشید که خروجی این متد قرار است توسط callBack یک متد جاوا اسکریپتی مورد استفاده قرار گرفته و احیانا عکس مورد نظر در صفحه نمایش داده شود. پس بهتر است که از قالب تصویر به شکل Base64 استفاده گردد. ضمن اینکه پلاگین‌های Jquery مرتبط با ویرایش تصویر هم از این قالب پشتیبانی می‌کنند. (اینجا)
این مثال به ساده‌ترین شکل نوشته شد. کلاس دیگری هم در اینجا وجود دارد و در صورتیکه از اسکنر نوع Feeder استفاده می‌کنید، می‌توانید از کدهای آن استفاده کنید.

کار ما تا اینجا در پروژه‌ی WCF Service تقریبا تمام است. اگر پروژه را یکبار Build نمایید برای اولین بار احتمالا پیغام خطاهای زیر ظاهر خواهند شد:

جهت رفع این خطا، در قسمت Referenceهای پروژه خود، WIA را انتخاب نموده و از Propertiesهای آن خصوصیت Embed Interop Types را به False تغییر دهید؛ مشکل حل می‌شود.

به سراغ پروژه‌ی ویندوز فرم جهت هاست کردن این WCF سرویس می‌رویم. می‌توانید این سرویس را بر روی یک Console App یا Windows Service هم هاست کنید که در اینجا برای سادگی مثال، از WinForm استفاده می‌کنیم.
یک پروژه‌ی WinForm جدید را ایجاد کنید و سپس از قسمت Solution > Add Reference به مسیر پروژه‌ی قبلی رفته و dllهای آن را به پروژه خود اضافه نمایید.
Form1.cs را باز کرده و ابتدا دو متغیر زیر را در آن به صورت عمومی تعریف نمایید:

```
private readonly Uri _baseAddress = new Uri("http://localhost:6019");
        private ServiceHost _host;
```

برای استفاده از کلاس ServiceHost لازم است تا ارجاعی به فضای نام System.ServiceModel داده شود. متغیر _baseAddress نگه دارنده‌ی آدرس ثابت سرویس اسکنر در سمت کلاینت می‌باشد و به این ترتیب ما دقیقا می‌دانیم باید سرویس را با کدام آدرس در کدهای جاوا اسکریپتی خود فراخوانی نماییم.
حال در رویداد Form_Load برنامه، کدهای زیر را جهت هاست کردن سرویس اضافه می‌نماییم:

```
private void Form1_Load(object sender, EventArgs e)
        {
            _host = new ServiceHost(typeof(WcfServiceScanner.ScannerService), _baseAddress);
            _host.Open();
        }

        private void Form1_FormClosing(object sender, FormClosingEventArgs e)
        {
            _host.Close();
        }
```

همین چند خط برای هاست کردن سرویس روی آدرس localhost و پورت 8010 کامپیوتر کلاینت کافی است. اما یکسری تنظیمات مربوط به خود سرویس هم وجود دارد که باید در زمان پیاده سازی سرویس، در خود پروژه‌ی سرویس، ایجاد می‌گردید. اما از آنجا که ما قرار است سرویس را در یک پروژه‌ی دیگر هاست کنیم، بنابراین این تنظیمات را باید در همین پروژه‌ی WinForm قرار دهیم.
فایل App.Config پروژه‌ی WinForm را باز کرده و کدهای آنرا مطابق زیر تغییر دهید:

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<configuration>
    <startup>
        <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" />
    </startup>

  <system.serviceModel>
    <behaviors>
      <serviceBehaviors>
        <behavior name="BehaviourMetaData">
          <serviceMetadata httpGetEnabled="true" />
        </behavior>
      </serviceBehaviors>
    </behaviors>
    <services>
      <service name="WcfServiceScanner.ScannerService"
               behaviorConfiguration="BehaviourMetaData">
        <endpoint address=""
                  binding="basicHttpBinding"
                  contract="WcfServiceScanner.IScannerService" />
      </service>
    </services>
  </system.serviceModel>

</configuration>
```

اکنون پروژه‌ی هاست آماده اجرا می‌باشد. اگر آنرا اجرا کنید و در مرورگر خود آدرس ذکر شده را وارد کنید، صفحه‌ی زیر را مشاهده خواهد کرد که به معنی صحت اجرای سرویس اسکنر می‌باشد.

ScannerService Service

You have created a service.

To test this service, you will need to create a client and use it to call the service. You can do this using the svcutil.exe tool from the command line with the following syntax:

```
svcutil.exe http://localhost:6019/?wsdl
```

You can also access the service description as a single file:

```
http://localhost:6019/?singleWsdl
```

This will generate a configuration file and a code file that contains the client class. Add the two files to your client application and use the generated client class to call the Service. For example:

C#

```
class Test
{
    static void Main()
    {
        ScannerServiceClient client = new ScannerServiceClient();

        // Use the 'client' variable to call operations on the service.

        // Always close the client.
        client.Close();
    }
}
```

اگر موفق به اجرا نشدید و احیانا با خطای زیر مواجه شدید، اطمینان حاصل کنید که ویژوال استودیو Run as Administrator باشد. مشکل حل خواهد شد.

به سراغ پروژه‌ی بعدی، یعنی وب سایت خود می‌رویم. یک پروژه‌ی MVC جدید ایجاد نمایید و در View مورد نظر خود، کدهای زیر را جهت صدا زدن متد GetScan اضافه می‌کنیم.

(از آنجا که کدها به صورت جاوا اسکریپت می‌باشد، پس مهم نیست که حتما پروژه MVC باشد؛ یک صفحه‌ی HTML ساده هم کافی است).

```
<a href="#" id="get-scan">Get Scan</a>
<img src="" id="img-scanned" />
<script>
    $("#get-scan").click(function () {
        var url = 'http://localhost:6019/';
        $.get(url, function (data) {
            $("#img-scanned").attr("src","  "+ data.GetScanResult);
        });
    });
</script>
```

دقت کنید در هنگام دریافت اطلاعات از سرویس، نتیجه به شکل GetScanResult خواهد بود. الان اگر پروژه را اجرا نمایید و روی لینک کلیک کنید، اسکنر شروع به دریافت اسکن خواهد کرد اما نتیجه‌ای بازگشت داده نخواهد شد و علت هم مشکل امنیتی CORS می‌باشد که به دلیل دریافت اطلاعات از یک دامین دیگر رخ می‌دهد و اگر با Firebug درخواست را بررسی کنید متوجه خطا به شکل زیر خواهید شد.

راه حل‌های زیادی برای این مشکل ارائه شده است، و متاسفانه بسیاری از آنها در شرایط پروژه‌ی ما جوابگو نمی‌باشد (به دلیل هاست روی یک پروژه ویندوزی). تنها راه حل مطمئن (تست شده) استفاده از یک کلاس سفارشی در پروژه‌ی WCF Service می‌باشد که مثال آن در اینجا آورده شده است.

برای رفع مشکل به پروژه WcfServiceScanner بازگشته و کلاس جدیدی را به نام CORSSupport ایجاد کرده و کدهای زیر را به آن اضافه کنید:

```
public class CORSSupport : IDispatchMessageInspector
    {
        Dictionary<string, string> requiredHeaders;
        public CORSSupport(Dictionary<string, string> headers)
        {
            requiredHeaders = headers ?? new Dictionary<string, string>();
        }

        public object AfterReceiveRequest(ref System.ServiceModel.Channels.Message request,
System.ServiceModel.IClientChannel channel, System.ServiceModel.InstanceContext instanceContext)
        {
            var httpRequest = request.Properties["httpRequest"] as HttpRequestMessageProperty;
            if (httpRequest.Method.ToLower() == "options")
                instanceContext.Abort();
            return httpRequest;
        }

        public void BeforeSendReply(ref System.ServiceModel.Channels.Message reply, object
correlationState)
        {
            var httpResponse = reply.Properties["httpResponse"] as HttpResponseMessageProperty;
            var httpRequest = correlationState as HttpRequestMessageProperty;

            foreach (var item in requiredHeaders)
            {
                httpResponse.Headers.Add(item.Key, item.Value);
            }
            var origin = httpRequest.Headers["origin"];
            if (origin != null)
                httpResponse.Headers.Add("Access-Control-Allow-Origin", origin);

            var method = httpRequest.Method;
            if (method.ToLower() == "options")
                httpResponse.StatusCode = System.Net.HttpStatusCode.NoContent;

        }
    }

    // Simply apply this attribute to a DataService-derived class to get
    // CORS support in that service
    [AttributeUsage(AttributeTargets.Class)]
    public class CORSSupportBehaviorAttribute : Attribute, IServiceBehavior
    {
        #region IServiceBehavior Members

        void IServiceBehavior.AddBindingParameters(ServiceDescription serviceDescription,
ServiceHostBase serviceHostBase, System.Collections.ObjectModel.Collection<ServiceEndpoint> endpoints,
BindingParameterCollection bindingParameters)
        {
        }

        void IServiceBehavior.ApplyDispatchBehavior(ServiceDescription serviceDescription,
ServiceHostBase serviceHostBase)
        {
            var requiredHeaders = new Dictionary<string, string>();

            //Chrome doesn't accept wildcards when authorization flag is true
```

```
            //requiredHeaders.Add("Access-Control-Allow-Origin", "*");
            requiredHeaders.Add("Access-Control-Request-Method", "POST,GET,PUT,DELETE,OPTIONS");
            requiredHeaders.Add("Access-Control-Allow-Headers", "Accept, Origin, Authorization, X-
Requested-With,Content-Type");
            requiredHeaders.Add("Access-Control-Allow-Credentials", "true");
            foreach (ChannelDispatcher cd in serviceHostBase.ChannelDispatchers)
            {
                foreach (EndpointDispatcher ed in cd.Endpoints)
                {
                    ed.DispatchRuntime.MessageInspectors.Add(new CORSSupport(requiredHeaders));
                }
            }
        }

        void IServiceBehavior.Validate(ServiceDescription serviceDescription, ServiceHostBase
serviceHostBase)
        {
        }

        #endregion
    }
```

فضاهای نام لازم برای این کلاس به شرح زیر می‌باشد:

```
using System.ServiceModel;
using System.ServiceModel.Channels;
using System.ServiceModel.Description;
using System.ServiceModel.Dispatcher;
```

کلاس ScannerService را باز کرده و آنرا به ویژگی

```
[CORSSupportBehavior]
    public class ScannerService : IScannerService
    {
```

مزین نمایید.

کار تمام است، یکبار دیگر ابتدا پروژه‌ی WcfServiecScanner و سپس پروژه هاست را Build کرده و برنامه‌ی هاست را اجرا کنید. اکنون مشاهده می‌کنید که با زدن دکمه‌ی اسکن، اسکنر فرم تنظیمات اسکن را نمایش می‌دهد که پس از زدن دکمه‌ی Scan، پروسه آغاز شده و پس از اتمام، تصویر اسکن شده در صفحه‌ی وب سایت نمایش داده می‌شود.

نظرات خوانندگان

نویسنده: امیر بختیاری
تاریخ: ۱۳۹۳/۱۱/۲۱ ۱:۴۸

با سلام و تشکر خدمت شما دوست گرامی
من تمام مراحل را به دقت ایجاد کردم ولی بازم همون خطایی که خودتون هم عنوان کردید را میده

```
Cross-Origin Request Blocked: The Same Origin Policy disallows reading the remote resource at
http://localhost:13748/. This can be fixed by moving the resource to the same domain or enabling CORS.
```

فایل پروژه را به طور کامل میزارم ببینید علت چیه WcfServiceScanner.zip

با تشکر مجدد

نویسنده: شروین ایرانی
تاریخ: ۱۳۹۳/۱۱/۲۱ ۹:۵

با سلام؛ آیا امکانش هست که سورس پروژه‌ی مثال رو هم قرار بدید؟ تشکر

نویسنده: امیر بختیاری
تاریخ: ۱۳۹۳/۱۱/۲۱ ۱۰:۳۱

سورس پروژه را از نظر قبلی می‌تونید دریافت کنید

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۱/۲۲ ۱۶:۱۵

این مثال را بررسی کنید: SelfHostWcfScannerService.zip
- ابتدا سرویس self host آن‌را با دسترسی ادمین اجرا کنید.
- بعد برنامه‌ی وب را اجرا کنید.

نویسنده: امیر بختیاری
تاریخ: ۱۳۹۳/۱۱/۲۲ ۱۷:۳۰

با تشکر
این مثال به قسمت Catch رفته و عکس پیشفرض رو نشان می‌دهد
درحالی که بر روی کامپیوتر من اسکنر نصب است و با دموی DynamSoft هم به خوبی جواب گرفتم و مشکلی نداشتم
به نظرتون چرا پنجره اسکنر رو باز نمیکنه. اسکنر مجازی میشه نصب کرد و تست گرفت

نویسنده: محمد رعیت پیشه
تاریخ: ۱۳۹۳/۱۱/۲۲ ۱۷:۵۲

اینجا رو بررسی کنید و ببینید خطایی که دریافت می‌کنید توضیحش چیست. موردی که خودم تست کردم این بود که وقتی روی سیستمی که اسکنر متصل نبود امتحان می‌کردم، به خطا می‌خوردم و به محض اتصال اسکنر خطا برطرف می‌شد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۱/۲۲ ۱۸:۱۷

در همان فایل ScannerService اگر catch را به این صورت تغییر دهید، استثناءها در کنسول سرویس نمایش داده می‌شوند.

```
catch(Exception ex)
{
  Console.WriteLine(ex);
```

برای مثال این مورد را من دریافت می‌کنم:

```
System.Runtime.InteropServices.COMException (0x80210015):
Exception from HRESULT: 0x80210015 at WIA.CommonDialogClass.ShowAcquireImage
```

کد 0x80210015 به معنای عدم یافت شدن اسکنر است (که صحیح است).

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۱/۲۲ ۱۸:۵۱

یک نکته‌ی تکمیلی

در متد GetScan در پایان کار، نیاز است منابع COM مورد استفاده رها شوند تا نشتی حافظه رخ ندهد:

```
finally
{
  if(image!=null) Marshal.ReleaseComObject(image);
  if(dialog!=null) Marshal.ReleaseComObject(dialog);
}
```

نویسنده: شروین ایرانی
تاریخ: ۱۳۹۳/۱۱/۲۸ ۱۰:۹

با سلام؛ من یه عکسی تهیه کردم از مراحل اجرای برنامه. ابتدا پروژه SelfHostService را Build کردم و سپس فایل exe را Run Administrator کردم و کنسول باز شد.سپس پروژه WebApplicationTest را بوسیله مرورگر فایرفاکس اجرا کردم. ولی تصویر پیش فرض را لود کرد. ضمنا در FireBug هم خطایی نداد. مشکل کجاست. اسکنر هم نصب هست. مرسی.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۱/۲۸ ۱۰:۱۳

در نظرات قبلی عنوان شد : «... در همان فایل ScannerService اگر catch را به این صورت تغییر دهید ...»

نویسنده: شروین ایرانی
تاریخ: ۱۳۹۳/۱۱/۲۸ ۱۰:۲۴

بله انجام شد. خطای 0x80210015 میده. ولی اسکنر کاملا نصب و کار میکنه در محیط‌های دیگه.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۱/۲۸ ۱۱:۸

- معنای دوم خطای 0x80210015 عدم سازگاری درایورهای اسکنر شما با WIA است (Windows Image Acquisition). اسکنر شما احتمالا فقط TWAIN compatible است و WIA or TWAIN compatible نیست.

- لیستی از این نوع اسکنرها

نویسنده: محمدرضابرخورداری
تاریخ: ۱۳۹۳/۱۲/۱۴ ۱۲:۱۱

با سلام؛ من تمامی مراحل رو طی کردم اما با خطای زیر مواجه شدم :

```
Host is running.
System.Runtime.InteropServices.COMException (0x80070050): The file exists. (Exception from HRESULT:
0x80070050)
at WIA.ImageFileClass.SaveFile(String Filename)
at SelfHost.ScannerService.GetScan() in
g:\project\asnad\scan\SelfHostWcfScannerService\SelfHostWcfService\ScannerService.cs:line 25
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۲/۱۴ ۱۲:۳۴

سطر Path.GetTempFileName را تبدیل کنید به مسیری که به آن دسترسی دارید. همچنین بررسی کنید نام این فایل منحصربفرد باشد.

نویسنده: محمدرضابرخوردار
تاریخ: ۱۳۹۳/۱۲/۱۶ ۱۹:۴۱

بسیار ممنون، مشکل حل شد. یه سوال دیگه : آیا در WIA امکان اسکن دو رو هم وجود داره؟ به چه شکلی

نویسنده: محمد رعیت پیشه
تاریخ: ۱۳۹۴/۰۱/۱۴ ۱۶:۴۲

پاسخ شما اسکن کردن در قالب تصاویر tiff هست، که می‌توانند چندین عکس در یک عکس باشند، مثل اسکن پشت و رو کارت ملی.
فقط برای نمایش در مرورگر به مشکل برخورد می‌کنید که یک راه استفاده از افزونه AlternaTIFF می باشد.

نویسنده: محمدرضابرخوردار
تاریخ: ۱۳۹۴/۰۱/۱۵ ۱۳:۱۲

Path.GetTempFileName باعث خطای The file exists می‌شود. زمانیکه با آدرس قابل دسترس و نام ثابت برای فایل جایگزین میکنم برای بار دوم همین خطا را میدهد.

نویسنده: محمد رعیت پیشه
تاریخ: ۱۳۹۴/۰۱/۱۶ ۲۳:۳۴

به نظرم یکتا بودن آدرس فایل در سمت کلاینت ضروری نمی‌باشد، در اینصورت می‌توانید یک مسیر ثابت و مشخص رو در نظر بگیرید و قبل از ذخیره، وجود اون رو بررسی کنید و چنانچه از قبل وجود داشت ابتدا آن را پاک نموده سپس ذخیره کنید.

عنوان: **تبدیل یک View به رشته و بازگشت آن به همراه نتایج JSON حاصل از یک عملیات Ajax ایی در ASP.NET MVC**
نویسنده: تایماز قبادی
تاریخ: ۱۳۹۳/۱۱/۲۳ ۱۸:۴۰
آدرس: www.dotnettips.info
گروه‌ها: MVC, JSON

ممکن است بخواهیم در پاسخ یک تقاضای Ajax ای ی، اگر عملیات در سمت سرور با موفقیت انجام شد، خروجی یک Controller action را به کاربر نهایی نشان دهیم. در چنین سناریویی لازم است که بتوانیم خروجی یک action را بصورت رشته برگردانیم. در این مقاله به این مسئله خواهیم پرداخت .
فرض کنید در یک سیستم وبلاگ ساده قصد داریم امکان کامنت گذاشتن بصورت Ajax را پیاده سازی کنیم. یک ایده عملی و کارآ این است: بعد از اینکه کاربر متن کامنت را وارد کرد و دکمه‌ی ارسال کامنت را زد، تقاضا به سمت سرور ارسال شود و اگر سرور پیغام موفقیت را صادر کرد، متن نوشته شده توسط کاربر را به کمک کدهای JavaScript و در همان سمت کلاینت بصورت یک کادر کامنت جدید به محتوای صفحه اضافه کنیم. بنده در اینجا برای اینکه بتوانم اصل موضوع مورد بحث را توضیح دهم، از یک سناریوی جایگزین استفاده می‌کنم؛ کاربر موقعیکه دکمه ارسال را زد، تقاضا به سرور ارسال میشود. سرور بعد از انجام عملیات، تحت یک شی JSON هم نتیجه‌ی انجام عملیات و هم محتوای HTML نمایش کامنت جدید در صفحه را به سمت کلاینت ارسال خواهد کرد و کلاینت در صورت موفقیت آمیز بودن عملیات، آن محتوا را به صفحه اضافه می‌کند.

با توجه به توضیحات داده شده، ابتدا یک شیء نیاز داریم تا بتوانیم توسط آن نتیجه‌ی عملیات Ajax ایی را بصورت JSON به سمت کلاینت ارسال کنیم:

```
public class MyJsonResult
{
  public bool success { set; get; }
  public bool HasWarning { set; get; }
  public string WarningMessage { set; get; }
  public int errorcode { set; get; }

  public string message {set; get; }
  public object data { set; get; }
 }
```

سپس به متدی نیاز داریم که کار تبدیل نتیجه‌ی action را به رشته، انجام دهد:

```
public static string RenderViewToString(ControllerContext context,
    string viewPath,
    object model = null,
    bool partial = false)
{
    ViewEngineResult viewEngineResult = null;
    if (partial) viewEngineResult = ViewEngines.Engines.FindPartialView(context, viewPath);
    else viewEngineResult = ViewEngines.Engines.FindView(context, viewPath, null);
    if (viewEngineResult == null) throw new FileNotFoundException("View cannot be found.");
    var view = viewEngineResult.View;
    context.Controller.ViewData.Model = model;
    string result = null;
    using(var sw = new StringWriter()) {
        var ctx = new ViewContext(context, view, context.Controller.ViewData,
context.Controller.TempData, sw);
        view.Render(ctx, sw);
        result = sw.ToString();
    }
    return result;
}
```

در اینجا موتور View را بر اساس اطلاعات یک View، مدل و سایر اطلاعات Context جاری کنترلر، وادار به تولید معادل رشته‌ای آن می‌کنیم.

فرض کنیم در سمت Controller هم از کدی شبیه به این استفاده میکنیم:

```
public JsonResult AddComment(CommentViewModel model) {
    MyJsonResult result = new MyJsonResult() {
        success = false;
    };
    if (!ModelState.IsValid) {
        result.success = false;
        result.message = "لطفاً اطلاعات فرم را کامل وارد کنید";
        return Json(result);
    }
    try {
        Comment theComment = model.toCommentModel();
        //EF service factory
        Factory.CommentService.Create(theComment);
        Factory.SaveChanges();
        result.data = Tools.RenderViewToString(this.ControllerContext, "/views/posts/_AComment", model,
true);
        result.success = true;
    } catch (Exception ex) {
        result.success = false;
        result.message = "اشکال زمان اجرا";
    }
    return Json(result);
}
```

و در سمت کلاینت برای ارسال Form به صورت Ajax ایی خواهیم داشت:

```
@using (Ajax.BeginForm("AddComment", "posts",
new AjaxOptions()
{
   HttpMethod = "Post",
   OnSuccess = "AddCommentSuccess",
   LoadingElementId = "AddCommentLoading"
}, new { id = "frmAddComment", @class = "form-horizontal" }))
{
    @Html.HiddenFor(m => m.PostId)
    <label for="fname">@Texts.ContactName</label>
    <input type="text" id="fname" name="FullName" class="form-control" placeholder="@Texts.ContactName
">
    <label for="email">@Texts.Email</label>
    <input type="email" id="InputEmail" name="email" class="form-control" placeholder="@Texts.Email">
    <br><textarea name="C_Content" cols="60" rows="10" class="form-control"></textarea><br>
    <input type="submit" value="@Texts.SubmitComments" name="" class="btn btn-primary">
    <div class="loading-mask" style="display:none">@Texts.LoadingMessage</div>
}
```

در اینجا در صورت موفقیت آمیز بودن عملیات، متد جاوا اسکریپتی AddCommentSuccess فراخوانی خواهد شد.
باید توجه شود Texts در اینجا یک Resource هست که به منظور نگهداری کلمات استفاده شده در سایت، برای زبانهای مختلف استفاده می‌شود (رجوع شود به مفهوم بومی سازی در Asp.net) .

و در قسمت script ها داریم:

```
<script type="text/javascript">
  function AddCommentSuccess(jsData) {
   if (jsData && jsData.message)
    alert(jsData.message);
   if (jsData && jsData.success) {
    document.getElementById("frmAddComment").reset();
      //افزودن کامنت جدید ساخته شده توسط کاربر به لیست کامنتهای صفحه
    $("#divAllComments").html(jsData.data + $("#divAllComments").html());
   }
  }
</script>
```

متد AddCommentSuccess اطلاعات شیء JSON بازگشتی از کنترلر را دریافت و سپس پیام آن‌را در صورت موفقیت آمیز بودن عملیات، به DIV ایی با id مساوی divAllComments اضافه می‌کند.

عنوان: **آشنایی با ساختار ViewBag**
نویسنده: احمد نواصری
تاریخ: ۱۳۹۳/۱۲/۰۴ ۱۵:۲۵
آدرس: www.dotnettips.info
گروه‌ها: C#, MVC, Dynamic

در روزهای اولی که با MVC آشنا شدم، این سؤال برایم پیش می‌آمد که یک ViewBag چطور می‌تواند به صورت پویا مقادیر را داخل خودش نگهداری کند؟ بعد از جستجو مشخص شد که ViewBag در حقیقت یک شیء Dynamic است. در این نوشتار قصد داریم نحوه‌ی کار یک ViewBag را نمایش دهیم. قبل از هر چیز باید بگویم که ViewBag تنها یک شیء dynamic نیست. اگر آن را از نوع dynamic تعریف و سپس یک شی را به آن Bind کنیم، در هنگام اجرای برنامه استثنای Cannot perform runtime binding صادر می‌شود. در حقیقت باید بگویم که ViewBag علاوه بر dynamic بودن، یک شیء از کلاس ExpandoObject است. با این تعاریف کلاس حاوی ViewBag ما بصورت زیر خواهد بود:

```
public class Controller
    {
        private dynamic _viewBage = new ExpandoObject();
        public  dynamic ViewBag
        {
            get { return _viewBage; }
        }
    }
```

حال برای استفاده از این ViewBag سفارشی کافی است تا کلاسی را تعریف کنیم که از کلاس پایه Controller ما ارث بری کند:

```
public class Sample : Controller
    {
        public  void ShowViewBag()
        {
            ViewBag.Title = 11;
            Console.WriteLine(ViewBag.Title);

            ViewBag.Title = "T";
            Console.WriteLine(ViewBag.Title);

            ViewBag.Title = false;
            Console.WriteLine(ViewBag.Title);

            ViewBag.Title = Math.PI;
            Console.WriteLine(ViewBag.Title);
        }
    }
}
```

در اینجا نحوه‌ی انتساب خواص پویا به ViewBag و مقدار دهی آن‌ها را مشاهده می‌کنید. همچنین در ذیل نحوه‌ی استفاده‌ی از آن‌را در یک برنامه‌ی کنسول بررسی کرده‌ایم:

```
class Program
    {
        static void Main()
        {
            Sample s = new Sample();
            s.ShowViewBag();
            Console.ReadKey();
        }
    }
```

خروجی حاصل از تکه کد بالا به صورت ذیل است:

file:///G:/Project/ViewBag/Vie
11
T
False
3.14159265358979

نظرات خوانندگان

نویسنده: نیکی
تاریخ: ۱۳۹۳/۱۲/۰۵ ۱۰:۳۸

کلمه dynamic چه کاری انجام میده؟ به عنوان نوع هست؟

نویسنده: احمد نواصری
تاریخ: ۱۳۹۳/۱۲/۰۵ ۱۲:۱۸

کلمه dynamic به کامپایلر اعلام میکنه که شی که تعریف کردیم میتونه تغییر کنه یعنی انواع مختلفی به خودش بگیره . شما وقتی یک شی رو با var تعریف میکنید و یک عدد صحیح به اون اختصاص میدید، در دفعه بعد نمیتونید چیزی غیر از عدد صحیح به اون اختصاص بدین. اما با استفاده از dynamic این کار امکان پذیره.

```
var x = 12; //OK
x = "String" //خطای کامپایل

dynamic y = 12; //OK
y="String"; //OK
y=12.7; //OK
```

عنوان: **مدیریت سشن‌ها در برنامه‌های وب به کمک تزریق وابستگی‌ها**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۲/۱۰ ۱۰:۴۵
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, Design patterns, MVC, Dependency Injection

سشن‌ها در برنامه‌های وب، یکی از وابستگی‌های استاتیکی هستند که می‌توان آن‌ها را از طریق تزریق وابستگی‌ها، جهت بالا بردن قابلیت آزمون پذیری برنامه، تامین کرد. همچنین اگر از سشن‌ها برای نمونه در برنامه‌های ASP.NET MVC استفاده کنید، مقدار آن‌ها در سازنده‌ی کنترلرها نال خواهند بود ؛ از این جهت که در زمان نمونه سازی یک کنترلر توسط IoC Container، کار مدیریت سشن‌ها صورت نمی‌گیرد و اگر در این بین سرویسی نیاز به سشن داشته باشد، دیگر وهله سازی نخواهد شد؛ به این دلیل که صرفنظر از مقدار دهی متغیر سشن در صفحه‌ای دیگر، این مقدار در سازنده‌ی کلاس، نال است. در ادامه این مشکل را از طریق غنی سازی تزریق وابستگی‌ها با اطلاعات سشن جاری، برطرف خواهیم کرد.

طراحی یک تامین کننده‌ی عمومی سشن

```
public interface ISessionProvider
{
    object Get(string key);
    T Get<T>(string key) where T : class;
    void Remove(string key);
    void RemoveAll();
    void Store(string key, object value);
}
```

در اینجا یک اینترفیس عمومی را مشاهده می‌کنید که کار آن کپسوله سازی اعمال متداول کار با سشن‌ها است؛ برای مثال دریافت اطلاعات یک سشن، بر اساس کلیدی مشخص و یا ذخیره سازی اطلاعات اشیاء در سشن‌ها.
یک نمونه پیاده سازی عمومی آن نیز برای کار با سشن‌ها در برنامه‌های وب ASP.NET وب فرم و MVC، می‌تواند به صورت زیر باشد:

```
public class DefaultWebSessionProvider : ISessionProvider
{
    private readonly HttpSessionStateBase _session;

    public DefaultWebSessionProvider(HttpSessionStateBase session)
    {
        _session = session;
    }

    public object Get(string key)
    {
        return _session[key];
    }

    public T Get<T>(string key) where T : class
    {
        return _session[key] as T;
    }

    public void Remove(string key)
    {
        _session.Remove(key);
    }

    public void RemoveAll()
    {
        _session.RemoveAll();
    }

    public void Store(string key, object value)
    {
        _session[key] = value;
    }
}
```

در کلاس DefaultWebSessionProvider مستقیما از HttpContext.Current.Session برای دسترسی به سشن جاری استفاده نشده‌است. این مقدار را از سازنده‌ی خود که توسط کلاس پایه HttpSessionStateBase تامین می‌شود، دریافت خواهد کرد. این سازنده را توسط تنظیمات ابتدایی IoC Container خود وهله سازی و مقدار دهی می‌کنیم؛ زیرا HttpContext.Current.Session برای مقدار دهی، نیاز به راه اندازی یک وب سرور دارد و عملا استفاده و شبیه سازی از آن در بسیاری از آزمون‌های واحد، بسیار مشکل خواهد بود.

```
private static Container defaultContainer()
{
    return new Container(ioc =>
    {
        // session manager setup
        ioc.For<ISessionProvider>().Use<DefaultWebSessionProvider>();
        ioc.For<HttpSessionStateBase>()
           .Use(ctx => new HttpSessionStateWrapper(HttpContext.Current.Session));

        ioc.Policies.SetAllProperties(properties =>
        {
            properties.OfType<ISessionProvider>();
        });
    });
}
```

در مثال فوق یک نمونه از تنظیمات ابتدایی StructureMap را برای استفاده از مقدار HttpContext.Current.Session، جهت وهله سازی سازنده‌ی کلاس DefaultWebSessionProvider مشاهده می‌کنید.

استفاده از تامین کننده‌ی سفارشی سشن در برنامه

پس از طراحی تامین کننده‌ی سفارشی سشن و همچنین معرفی آن به IoC Container خود، اکنون استفاده‌ی از آن به سادگی ذیل است:

```
public class HomeController : Controller
{
    private readonly ISessionProvider _sessionProvider;
    public HomeController(ISessionProvider sessionProvider)
    {
        _sessionProvider = sessionProvider;
    }
```

بنابراین اگر در کلاسی، کنترلری و یا سرویسی نیاز به سشن وجود داشت، بهتر است از ISessionProvider بجای مقدار دهی و یا دسترسی مستقیم به شیء استاتیک Session استفاده کرد.

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۲/۱۴ ۱۵:۲۵

یک نکته‌ی تکمیلی

شبیه به سشن، چند مورد دیگر را نیز می‌توان به همین روش تامین کرد:

```
ioc.For<IIdentity>().Use(() => HttpContext.Current.User.Identity);
ioc.For<HttpContextBase>().Use(() => new HttpContextWrapper(HttpContext.Current));
ioc.For<HttpServerUtilityBase>().Use(() => new HttpServerUtilityWrapper(HttpContext.Current.Server));
ioc.For<HttpRequestBase>().Use(ctx => ctx.GetInstance<HttpContextBase>().Request);
```

عنوان: **خلاصه‌ای از روش‌های ارسال داده‌های سمت سرور به کدهای جاوا اسکریپتی در ASP.NET MVC**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۱۲/۱۳ ۱۳:۵
آدرس: www.dotnettips.info
گروه‌ها: JavaScript, MVC, JSON

روش‌های زیادی برای ارسال داده‌های سمت سرور تهیه شده در یک برنامه‌ی ASP.NET به کدهای سمت کاربر JavaScript ایی وجود دارند که تعدادی از مهم‌ترین‌های آن‌ها را در این مطلب بررسی خواهیم کرد.

روش اول: دریافت اطلاعات سمت سرور به کمک درخواست‌های Ajax

استفاده از Ajax یکی از روش‌های کلاسیک دریافت اطلاعات سمت سرور در کدهای جاوا اسکریپتی است.

```
<script type="text/javascript">
var products = [];
$(function() {
    $.getJSON("/home/products", function(response) {
        products = response.products;
    });
});
</script>
```

برای نمونه در اینجا با استفاده از امکانات jQuery، درخواست Ajax ایی به سرور ارسال شده و سپس نتیجه‌ی دریافتی، به آرایه‌ی جاوا اسکریپتی products نسبت داده شده‌است.
- مزایا: استفاده از Ajax، روشی بسیار متداول و شناخته شده‌است و به کمک انواع و اقسام روش‌های بازگشت JSON از سرور، می‌توان با آن کار کرد.
- معایب: درخواست Ajax، صرفا پس از بارگذاری اولیه‌ی صفحه به سمت سرور ارسال خواهد شد و در این بین، کاربر وقفه‌ای را مشاهده خواهد کرد. همچنین در اینجا بجای یک درخواست از سرور، حداقل دو درخواست باید ارسال شوند؛ یکی برای بارگذاری صفحه‌ی اصلی و دیگری برای دریافت اطلاعات Ajax ایی از سرور به صورت غیرهمزمان.

روش دوم: دریافت اطلاعات از یک فایل جاوا اسکریپتی خارجی

اطلاعات سمت کاربر را از یک فایل جاوا اسکریپتی خارجی الحاق شده‌ی به صفحه‌ی جاری نیز می‌توان تهیه کرد:

```
<script src="/file.js"></script>
```

یک چنین فایلی را می‌توان توسط کدهای سمت سرور نیز بازگشت داد و اطلاعات آن‌را تهیه و پر کرد. در این حالت فقط کافی است که ContentType شیء Response را از نوع application/javascript مشخص کنیم و سپس یک خروجی متنی جاوا اسکریپتی را از سمت سرور بازگشت دهیم.
این روش نیز تقریبا مانند حالت یک درخواست Ajax ایی کار می‌کند و اطلاعات مورد نیاز را در طی یک درخواست جداگانه، پس از بارگذاری صفحه‌ی اصلی، از سرور دریافت خواهد کرد. البته در حالت کار با Ajax، می‌توان در طی یک callback، نتیجه را دریافت کرد و سپس عکس العمل نشان داد؛ اما در اینجا callback ایی وجود ندارد.

روش سوم: استفاده از SignalR

در SignalR ابتدا سعی می‌شود تا با استفاده از Web Sockets ارتباطی ماندگار بین کلاینت و سرور برقرار شود و سپس در این حالت، سرور می‌تواند مدام اطلاعاتی، مانند تغییرات داده‌های خود را به سمت کاربر، جهت نمایش و یا محاسبات خاص خود ارسال کند. اگر حالت Web Socket میسر نباشد (توسط سرور یا کلاینت پشتیبانی نشود)، به حالت‌های دیگری مانند server events, forever frames, long polling سوئیچ خواهد کرد. اطلاعات بیشتر

روش چهارم: قرار دادن اطلاعات سمت سرور در کدهای HTML صفحه

روش متداول دیگری جهت تامین اطلاعات جاوا اسکریپتی سمت کاربر، قرار دادن آن‌ها در ویژگی‌های data-* ارائه شده در HTML5 است.

```
<ul>
@foreach (var product in products)
{
  <li id="product@product.Id" data-rank="@product.Rank">@product.Name</li>
}
</ul>
```

در اینجا لیستی از محصولات، به صورت متداولی از کنترلر دریافت گردیده و سپس ویژگی data-rank هر ردیف نمایش داده شده نیز توسط این اطلاعات سمت سرور مقدار دهی شده‌اند.

اکنون برای دسترسی به مقدار data-rank سطری مانند product1، در کدهای جاوا اسکریپتی صفحه می‌توان نوشت:

```
<script type="text/javascript">
var product1Rank = $("#product1").data("rank");
</script>
```

این روش برای قرار دادن اطلاعات ثابت اشیاء، روش مرسومی است.

روش پنجم: قرار دادن اطلاعات سمت سرور در کدهای جاوا اسکریپتی صفحه

این روش همانند روش چهارم است، با این تفاوت که اینبار اطلاعات مورد نیاز، مستقیما به یک متغیر جاوا اسکریپتی انتساب داده شده‌است:

```
<script type="text/javascript">
var product1Name = "@product1.Name";
</script>
```

مزیت این روش، عدم ارسال درخواست اضافه‌تری به سرور برای دریافت اطلاعات مورد نیاز است. مقدار product1Name در همان بار اول رندر صفحه از سمت سرور، مشخص می‌گردد.

روش ششم: انتساب یک شیء دات نتی به یک متغیر جاوا اسکریپتی

این روش همانند روش پنجم است، با این تفاوت که اینبار قصد داریم بجای یک مقدار ثابت رشته‌ای یا عددی، برای مثال، آرایه‌ای از اشیاء را به یک متغیر جاوا اسکریپتی انتساب دهیم. در اینجا ابتدا اطلاعات مورد نظر را به فرمت JSON تبدیل می‌کنیم:

```
//سمت سرور
[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();
}
```

سپس توسط متد Html.Raw می‌توان این رشته‌ی JSON را که اکنون حاوی آرایه جاوا اسکریپتی سمت سرور است، به یک متغیر جاوا اسکریپتی نسبت داد:

```
//سمت کلاینت
<script type="text/javascript">
var jsonArray = @Html.Raw(@ViewBag.JsonString);
</script>
```

استفاده از Html.Raw در این حالت از این جهت ضروری است که اطلاعاتی مانند [] به صورت encode شده در سمت کاربر نمایش داده نشوند؛ چون Razor به صورت پیش فرض اطلاعات را Encode می‌کند.
و یا اینکار را به صورت خلاصه به شکل زیر نیز می‌توان در سمت کاربر انجام داد:

```
<script type="text/javascript">
  var model = @Html.Raw(Json.Encode(Model));
  // your js code here
</script>
```

در اینجا کار تبدیل اطلاعات مدل به JSON، در همان سمت RazorView انجام شده‌است.

عنوان: یکدست کردن "ی" و "ک" در ASP.NET MVC با پیاده‌سازی یک Model Binder
نویسنده: محسن درپرستی
تاریخ: ۱۳۹۴/۰۱/۰۴ ۱:۳۵
آدرس: www.dotnettips.info
گروه‌ها: MVC, Persian

قبلا در همین وب‌سایت در مورد یکسان‌سازی حروف "ی" و "ک" مطلبی بیان شده است. تمرکز آن مطلب بر روی اعمال تغییرات، قبل از ذخیره در دیتابیس با استفاده از EF است. به عبارتی، متن‌هایی که توسط مدیر سایت یا هر کاربر دیگری برای ذخیره شدن در دیتابیس وارد شده است. اما در صورتی که کاربری در جستجوی خود از حروف "ی" و "ک" عربی استفاده کند چه می‌شود؟ در اینجا می‌خواهیم روشی را پیاده‌سازی کنیم که عمومی بوده و بتواند هر دوی این حالات را پوشش دهد.

مسئله این است که بایستی تمام ورودی‌های کاربران سایت که از نوع رشته هستند چک و در صورت نیاز اصلاح شوند. بهترین روشی که به ذهن من می‌رسد این است که در فرایند Model Binding این عمل انجام بگیرد. با تعریف یک Model Binder برای نوع رشته می‌توانیم عمل یکدست‌سازی را به صورت عمومی در تمام وب‌سایت اعمال کنیم.

در MVC یک Model Binder پیش‌فرض داریم به نام DefaultModelBinder . ما هم از همین کلاس استفاده می‌کنیم تا تمام کارها را برای ما انجام دهد. تنها بایستی در متد BindModel آن کد موردنظر خود را اضافه کنیم و سپس اجازه دهیم بقیه فرایند به شکل عادی ادامه پیدا کند.

کلاسی به نام StringModelBinder اضافه کرده و کلاس DefaultModelBinder را به عنوان کلاس پایه آن تعریف می‌کنیم. سپس متد BindModel آنرا override کرده، کد مربوط به یکدست‌سازی را اضافه می‌کنیم :

```
public class StringModelBinder : DefaultModelBinder
    {
        public override object BindModel(ControllerContext controllerContext, ModelBindingContext
bindingContext)
        {
            object value = base.BindModel(controllerContext, bindingContext);

            if (value == null)
            {
                return value;
            }

            return value.ToString().Replace((char)1610, (char)1740).Replace((char)1603, (char)1705);
        }
    }
```

ابتدا مقدار به دست آمده توسط متد پیش‌فرض را می‌گیریم. سپس در صورت نیاز یکسان‌سازی را انجام می‌دهیم.

برای فعال کردن این Model Binder بایستی آنرا در رویداد Application_start برنامه، برای نوع رشته به ModelBinder های برنامه اضافه کنیم :

```
ModelBinders.Binders.Add(typeof(string), new StringModelBinder());
```

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۴/۰۱/۰۶ ۲۱:۴۱

در مطلب تکمیلی « یک دست سازی ی و ک در برنامه‌های Entity framework 6 » روش دیگری برای اینکار معرفی شده‌است. در این حالت تمام کوئری‌هایی که توسط EF صادر می‌شوند و تمام پارامترهای آن‌ها پیش از ارسال به بانک اطلاعاتی، تحت کنترل قرار می‌گیرند (هر دو حالت کوئری‌های select و یا insert/update/delete توسط interceptorها در اختیار هستند و نه فقط حالت insert/update/delete مطلب قبلی).

عنوان: **حذف پردازش درخواست‌های فایل‌های استاتیک در متد Application_AuthenticateRequest**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۴/۰۱/۰۷ ۱۶:۵
آدرس: www.dotnettips.info
گروه‌ها: Entity framework, MVC, DNTProfiler

پروژه‌ی « فروشگاه شهر طلایی من » را اگر در برنامه‌ی DNTProfiler بررسی کنیم، در برگه‌ی Urls آن یک چنین گزارش‌هایی را می‌توان مشاهده کرد:

همانطور که مشاهده می‌کنید، درخواست یک فایل استاتیک، سبب اجرای یک کوئری بر روی بانک اطلاعاتی شده‌است و یک Context خاص خودش را نیز ایجاد کرده‌است. اگر به قسمت سابقه‌ی متدهایی که سبب بروز این امر شده‌اند (در همان برگه، در پایین صفحه) دقت کنیم، به متد Application_AuthenticateRequest فایل global.asax.cs می‌رسیم. هر چند در فایل RouteConfig.cs مسیرهای اسکریپت‌ها و فایل‌های CSS جهت صرفنظر شدن معرفی شده‌اند، اما این موارد بر روی متد خاص Application_AuthenticateRequest تاثیری ندارند و این متد به ازای هر درخواست رسیده‌ی به IIS یکبار اجرا می‌شود؛ زیرا

یک چنین تنظیمی در فایل web.config وجود دارد:

```
<modules runAllManagedModulesForAllRequests="true">
```

به همین دلیل است که حتی درخواست فایل‌های استاتیک نیز سبب اجرای ماژول forms authentication و بروز صدور یک کوئری شده‌اند.

کنترل ASP.NET MVC Bundles در حین Forms Authentication

اگر بخواهیم درخواست‌های رسیده‌ی به Application_AuthenticateRequest را کنترل کنیم، می‌توان چنین متدی را تدارک دید:

```
private bool shouldIgnoreRequest()
{
    string[] reservedPath =
    {
        "/__browserLink",
        "/img",
        "/fonts",
        "/Scripts",
        "/Content"
    };

    var rawUrl = Context.Request.RawUrl;
    if (reservedPath.Any(path => rawUrl.StartsWith(path, StringComparison.OrdinalIgnoreCase)))
    {
        return true;
    }

    return BundleTable.Bundles.Select(bundle => bundle.Path.TrimStart('~'))
              .Any(bundlePath => rawUrl.StartsWith(bundlePath, StringComparison.OrdinalIgnoreCase));
}
```

در اینجا یک سری مسیر مشخص مانند پوشه‌ی تصاویر، قلم‌ها و اسکریپت‌ها و امثال آن معرفی شده‌اند. همچنین BundleTable.Bundles نیز مورد بررسی قرار گرفته‌است. در حین استفاده‌ی از ASP.NET MVC Bundles دیگر مسیرها الزاما به پوشه‌ی Content ختم نخواهند شد:

```
bundles.Add(new ScriptBundle("~/bundles/modernizr").Include(
 "~/Scripts/modernizr-*"));
```

و بر اساس تعریف پارامتر ~/bundles/modernizer مسیری مانند localhost/bundles/modernizer درخواست خواهد شد. برای دسترسی به این مسیرهای سفارشی تعریف شده می‌توان از مجموعه‌ی BundleTable.Bundles، مطابق متد فوق کوئری گرفت و مسیرهای درخواستی را که با مسیرهای bundles سفارشی تعریف شده تطابق دارند، دیگر پردازش نکرد:

```
protected void Application_AuthenticateRequest(Object sender, EventArgs e)
{
    if (shouldIgnoreRequest()) return;

    if (Context.User == null)
        return;
```

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۴/۰۱/۱۳ ۰:۲۲

یک نکته‌ی تکمیلی

گزارش گیری از یک چنین درخواست‌هایی را در افزونه‌ی static resources queries نیز می‌توانید مشاهده کنید:

عنوان: **دریافت خطاهای موجود در Viewهای ASP.NET MVC در زمان کامپایل**
نویسنده: احمد رجبی
تاریخ: ۱۳۹۴/۰۱/۱۳ ۱۸:۵۵
آدرس: www.dotnettips.info
گروه‌ها: Debugging, MVC

روشهای زیادی برای انجام این کار وجود دارد:

1- در فایل پروژه (Your-MVC-Project.csproj) مقدار تگ MvcBuildViews را به true تغییر دهید.

2- استفاده از RazorGenerator

3- اگر Resharper نصب شده است، روشن کردن Solution-wide analysis (گوشه‌ی پایین سمت راست ویژوال استودیو) همه‌ی خطاهای پروژه از جمله خطاهای موجود در ویوها را نمایش خواهد داد.

4- و روشهای دیگر ...

اشکال روش اول، در طولانی شدن زمان کامپایل است و در روش دوم باید از یک کتابخانه جانبی برای این کار استفاده کنیم (اگر صرفا بخواهیم فقط از این قابلیت استفاده کنیم) و روش سوم را هم خودتان می‌توانید حدس بزنید! (مصرف بیش از حد منابع سیستم)

یک راه ساده‌تر این است که از روش اول فقط در زمانیکه پروژه در حالت Release قرار دارد، استفاده کنیم. در این صورت ویوها در حالت Debug کامپایل نمی‌شوند و برای کامپایل باید به حالت Release سوئیچ کنیم.
برای این کار مراحل زیر را انجام دهید:
1- کلیک راست بر روی پروژه و انتخاب Unload Project.
2- کلیک راست بر روی پروژه و انتخاب Edit Your-MVC-Project.csproj.
3- پیدا کردن تگ <MvcBuildViews>...</MvcBuildViews> و حذف آن.
4- افزودن کدهای زیر به فایل:

```
<PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU'">
    <MvcBuildViews>true</MvcBuildViews>
  </PropertyGroup>
```

به این صورت:

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <Project ToolsVersion="12.0" DefaultTargets="Build" xmlns="http://schemas.microsoft.c
3    <Import Project="$(MSBuildExtensionsPath)\$(MSBuildToolsVersion)\Microsoft.Common.p
4
5    <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU'">
6      <MvcBuildViews>true</MvcBuildViews>
7    </PropertyGroup>
8
9    <PropertyGroup>
```

5- ذخیره‌ی فایل، راست کلیک بر روی پروژه و انتخاب Reload Project.

اکنون پروژه‌ی خود را در حالت Release قرار داده و Build کنید!