

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 تابلحال چندین view engine، routing engine و غیره توسط برنامه نویس‌های مستقل برای ASP.NET MVC طراحی شده‌اند که هیچکدام با ASP.NET web forms میسر نیست. برای مثال از view engine پیش فرض آن خوششان نمی‌آید؟ عوضش کنید ! سیستم اعتبار سنجی توکار آن را دوست ندارید؟ آن را با یک نمونه بهتر تعویض کنید و الی آخر ...

به علاوه طراحی بر اساس 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 در نظر 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 کاملاً 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 عموماً [مشکل ساز بود](#). البته ASP.NET web forms در نگارش‌های جدید خود مشکل عدم امکان مقدار دهی ClientId سفارشی را برای کنترل‌های وب خود برطرف کرده است و این مورد را می‌توان دستی هم تنظیم کرد ولی در کل باز هم آنچنان کنترلی رو خروجی HTML نهایی کنترل‌های تولیدی نیست مگر اینکه مانند مورد چهارم یاد شده یک کنترل را از صفر بازنویسی کنید! همچنین اگر باز هم بیشتر با jQuery و ASP.NET web forms کار کرده باشید می‌دانید که jQuery آنچنان سنجی با ViewState و Postback وب فرم‌ها ندارد و همین مساله عموماً مشکل‌زا است. علاوه بر آن اخیراً میکروسافت توسعه ASP.NET Ajax خود را تقریباً 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 نت

طراحی اصلی ASP.NET web forms مربوط است به دوران دات نت یک؛ زمانیکه نه Generics وجود داشت، نه LINQ و نه آنچنان مباحث TDD یا استفاده از ORMs متداول بود. برای مثال شاید ایجاد یک strongly typed web form الان کمی دور از ذهن به نظر برسد، زمانیکه اصل آن بر مبنای بکارگیری گسترده datatable و dataset بوده است (با توجه به امکانات زبان‌های دات نت در آن دوران). بنابراین اگر علاقمند هستید که این امکانات جدید را بکار بگیرید، 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 کنترل‌ها چگونه باید تعیین شوند و قرار داد مرتبط به آن چیست، مدل باید کجا قرار گیرد، قرار داد پردازش آدرس‌های صفحات سایت به چه نحوی است و 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 میکروسافت نیست

پیاده سازی‌های مستقلی هم از ASP.NET MVC توسط اشخاص و گروه‌های خارج از میکروسافت 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 `url helper`, `html helper` و `ajax helper` ساده را خواهید دید؛ این نقطه ضعف ASP.NET MVC نیست. عمداً 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 شکل 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 و بلاگ‌ها 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: ۱۱:۵۹:۱۸ ۱۳۹۱/۰۱/۰۳

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: ۱۴:۲۶:۴۵ ۱۳۹۱/۰۱/۰۳

سلام و تبریک سال نو خدمت جناب آقای 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: ۱۴:۳۶:۲۳ ۱۳۹۱/۰۱/۰۳

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

نویسنده: شاهین کیاست

تاریخ: ۱۶:۱۰:۵۷ ۱۳۹۱/۰۱/۰۳

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: ۱۶:۳۰:۴۷ ۱۳۹۱/۰۱/۰۳

سلام خوشحالم که شروع کردید.

نویسنده: امین

تاریخ: ۲۳:۴۷:۰۴ ۱۳۹۱/۰۱/۰۳

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: ۰۰:۵۳:۲۶ ۱۳۹۱/۰۱/۰۴

کل سیستم WebAccess در TFS 11 از ابتدا با ASP.NET MVC پیاده سازی شده که واقعاً تمیز و سریع و جامع است. قابل مقایسه با نسخه های 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 بدهد؟ 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 تقریباً 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 اولاً که این مدارک 2 سال پیش تنظیم شده و ثانیاً 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 قبلاً با MVC با جوملا در ساخت مازول 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 Validation 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، شاید من اشتباه میکنم. نظر شما چیست؟

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

- هر وقت یادگرفتید («همه» منظورم هستند! نه شخص خاصی)، این تجربیات کوچک خودتون رو در قالب یک مطلب وبلاگ شخصی مستند کنید، اون وقت می بینید که این روزها شاید شد ساعت یا دقیقه برای نفر بعدی.
- شما در web form پیش نمایش خاص و آنچنانی دارید؟ مثلاً 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 می گیرد. شما با یک سیستم پویا سروکار دارید و نه با یک HTML ثابت.

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

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 نداریا!!!!!! (: (: (: {

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

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

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

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

نویسنده: amir
تاریخ: ۱۸:۲۷ ۱۳۹۱/۰۵/۱۴

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 می بینی 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 می خواستم ببینم کسی این کارو کرده . به نتیجه ای رسیده ؟ جواب میده ؟

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

dataset مربوط به دات نت 2 است. لازم است یک سری پیش نیاز رو اول مطالعه کنید:

[وضعیت فناوری های مرتبط با دات نت از دیدگاه مرگ و زندگی!](#)

[5 دلیل برای استفاده از یک ابزار ORM](#)

[امنیت در LINQ to SQL](#) (برای مابقی ORM ها هم به همین صورت است)

[مروری بر کدهای کلاس SqlHelper](#)

نویسنده: علی
تاریخ: ۱۴:۴۰ ۱۳۹۱/۰۵/۱۸

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

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

دریافت تمام مثال های MVC این سری:

[MVC_Samples.zip](#)

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

سلام

خیلی مقایسه مفید و کاملی بود ..

من حتی با خواندن کتاب Wrox Pro Asp.Net MVC 4 هم تمام این اطلاعات را پیدا نکردم

ممنون از سایت خوبتان (:

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

آیا به غیر از در نظر گرفتن حذف ViewState در برنامه سرعت نمایش صفحات در Asp.net 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: ۱۳:۳۹ ۱۳۹۲/۰۸/۰۹

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

با سرعت خوبی مواجه خواهید بود)

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

نویسنده: امیر خلیلی

تاریخ: ۱۶:۳۹ ۱۳۹۲/۰۸/۰۹

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

نویسنده: وحید نصیری

تاریخ: ۱۶:۵۴ ۱۳۹۲/۰۸/۰۹

فقط در زمان اولین بارگذاری برنامه در حافظه و اجرای اولیه آن.
برای این مساله هم در IIS های 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: ۱۶:۲۵ ۱۳۹۲/۰۹/۱۴

با سلام

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

نویسنده: وحید نصیری

تاریخ: ۱۸:۱۴ ۱۳۹۲/۰۹/۱۴

- 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. مثلاً [twitter bootstrap 3](#) یک فریم ورک CSS اصطلاحاً mobile first است. یعنی طوری طراحی شده که سایت شما را به خوبی بتواند با اندازه‌های کوچک نمایشگرها تطابق دهد و قابل استفاده کند.

نتیجه گیری؟

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