عنوان: **چگونه نرم افزارهای تحت وب سریعتری داشته باشیم؟ قسمت اول**
نویسنده: م منفرد
تاریخ: ۱۳۹۲/۰۵/۰۹ ۱۸:۰
آدرس: www.dotnettips.info
برچسب‌ها: ASP.Net, Performance

در این سلسله مقالات قصد دارم چندین مطلب راجع به افزایش سرعت نرم افزارهای تحت وب مطرح نمایم. این مطالب هرچند بسیار مختصر می‌باشند ولی در کارایی و سرعت برنامه‌های شما در آینده تاثیر خواهند داشت.

1.کش کردن همیشه آخرین حربه می‌باشد

این مهم است که بخش‌های مختلف سایت شما در سطوح مختلف کش شوند (ASP.NET, Kernel, Server, Proxy Server, Browser ,...) ولی این موضوع باید همیشه آخرین حربه و نکته ای باشد که آن را در مورد سایت خود اعمال می‌کنید.
یعنی همیشه مطمئن شوید ابتدا تمامی نکات مربوط به افزایش کارایی در برنامه خود را رعایت کرده اید، سپس اقدام به کش داده‌ها در سطوح مختلف نمایید. توجه کنید کش کردن داده‌ها و صفحات می‌تواند مشکلات را برای شما به عنوان یک برنامه نویس یا تست کننده برنامه پنهان کند و به شما اطمینان دهد که همه چیز خوب کار می‌کند در حالی که این چنین نیست!
البته ذکر این نکته هم بی فایده نیست که کش کردن همه چیز بعضی مواقع دشمن برنامه شما محسوب می‌شود! هیچ وقت یادم نمی‌رود، در پورتال داخلی یک شرکت که در وقت استراحت به کارکنان اجازه مطالعه روزنامه‌های روز را می‌داد (به صورت آفلاین)، این نکته در بالای صفحه آورده شده بود: «لطفا برای به روز رساندن صفحات روزنامه‌ها از کلید Ctrl+F5 استفاده نمایید». این موضوع یعنی بحث کشینگ در برنامه آن پرتال در سطح فاجعه می‌باشد! حالا فرض کنید این مشکل در فرم ورود و یا مرور اطلاعات یک برنامه به وجود آید...

2.حذف View Engineهای غیر ضروری

به عنوان یک برنامه نویس ASP.NET MVC، باید اطلاع داشته باشید که CLR به صورت خودکار View Engineهای Razor و Web Forms را لود می‌کند. این موضوع به این دلیل است که اطلاعی از نحوه برنامه نویسی شما ندارد. اگر شما فقط از یکی از این دو View Engine استفاده می‌کنید،لطفا دیگری را غیر فعال کنید! فعال بودن هر دوی آنها یعنی اتلاف وقت گرانبهای CPU سرور شما برای رندر کردن تمامی صفحات شما توسط دو انجین! ایتدا viewهای شما با Web Forms Engine رندر شده سپس نتیجه به Razor Engine منتقل شده و مجدد توسط این انجین رندر می‌شود. این موضوع در سایت‌های با تعداد کاربر بالا یعنی فاجعه!
برای حل این مشکل کافی است خطوط زیر را در فایل Global.asax و در رویداد بخش Application_Start وارد نمایید:

```
ViewEngines.Engines.Clear();
ViewEngines.Engines.Add(new RazorViewEngine());
```

این دو خط یعنی خداحافظ Web Forms Engine...
قبل از استفاده از این کد، اطمینان حاصل کنید کل برنامه شما توسط Razor تهیه شده است وگرنه بنده هیچ مسئولیتی در رابطه با فریادهای کارفرمای شما متقبل نمی‌شوم!
صد البته برای حذف Razor Engine و استفاده از Web Form Engine می‌توان از کد زیر در همان موقعیت فوق استفاده کرد:

```
ViewEngines.Engines.Clear();
ViewEngines.Engines.Add(new WebFormViewEngine());
```

البته همانطور که حتما دوستان مطلع هستند امکان گسترش Engineهای فوق توسط ارث بری از کلاس BuildManagerViewEngine جهت ایجاد Engineهای دیگر همیشه محیا است. در این صورت می‌توانید تنها انجین سفارشی مورد نظر خود را لود کرده و از لود دیگر انجین‌ها پرهیز کنید.

3. استفاده از فایل‌های PDB مایکروسافت برای دیباگ و یا پروفایل کردن DLLهای دیگران

دیباگ یا پروفایل کردن برنامه ها، DLL ها، اسمبلی‌ها و منابعی از برنامه که شما آن را خود ننوشته اید (سورس آنها در دسترس شما نمی‌باشد) همیشه یکی از سخت‌ترین مراحل کار می‌باشد. جهت کمک به دیباگرها یا پروفایلرها، نیاز است فایل‌های PDB مرتبط با DLLها را در اختیار آنها قرار دهید تا به بهترین نتیجه دسترسی پیدا کنید. این فایل‌ها محتوی نام توابع، شماره خطوط برنامه و metadataهای دیگر برنامه اصلی قبل از optimize شدن توسط کامپایلر یا JIT می‌باشد. خوب حالا اگر نیاز شد این کار را در رابطه با DLLها و کلاس‌های پایه Microsoft.NET انجام دهیم چه کار کنیم؟

خیلی ساده! خود Microsoft سروری جهت این موضوع تدارک دیده که فایل‌های PDB را جهت دیباگ کردن در اختیار تیم‌های برنامه نویسی قرار می‌دهد.کافی است از منوی Tools گزینه Options را انتخاب، سپس به بخش Debugging و به بخش Symbols بروید و گزینه Microsoft Symbol Servers as your source for Symbols را انتخاب کنید. برای اطمینان از اینکه هر مرتبه که برنامه را دیباگ می‌کنید مجبور به دانلود این فایل‌ها نشوید، فراموش نکنید پوشه ای را جهت کش این فایل‌ها ایجاد و آدرس آن را در بخش Cache symbols in this directory همین صفحه وارد نمایید.

این امکان در Visual Studio 2010, 2012 در دسترس می‌باشد.

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۵/۱۰ ۱۷:۱۴

اثر View Engineهای اضافی رو با Glimpse بهتر میشه دید.