

WinRT چیست؟

مایکروسافت جهت سهولت تولید برنامه‌های جدید Metro-style، لمسی (touch-centric) و tablets و ویندوز 8، اقدام به بازنویسی مجدد Windows-API کرده و نام آن را WinRT گذاشته است. بنابراین همانند آنچه که در مورد API ویندوز از روز اول پیدایش آن مرسوم بوده، این API جدید، از نوع native (نوشته شده با CPP) می‌باشد و با کمک دات نت فریم ورک تهیه نشده است. این API جدید مبتنی بر فناوری قدیمی COM است، بنابراین مطابق معمول توسط هر نوع برنامه و سیستمی در ویندوز قابل دسترسی است. تفاوتی نمی‌کند که CPP یا دلفی باشد یا دات نت. به صورت خلاصه ویندوز 8 دو طراحی جدید (WinRT) و قدیم (Win32 API) را با هم پشتیبانی می‌کند. اگر آن‌را صحیح‌تر بخواهیم معرفی کنیم، WinRT درحقیقت محصور کننده‌ی (Wrapper) همان Win32 API سابق است (در پشت صحنه همان dll های سابق ویندوز را بارگذاری و استفاده می‌کند) جهت تطابق با نیازهای دهه اخیر و سال‌های پیش رو.

سازگاری دات نت فریم ورک با WinRT چگونه است؟

اینبار WinRT برخلاف Win32 API (که در زمان ارائه آن اصلاً دات نتی در کار نبود)، جهت سازگاری با دات نت طراحی شده است. این طراحی جدید ILDasm metadata را در اختیار برنامه نویسی‌های دات نت قرار می‌دهد و به این ترتیب IntelliSense و قابلیت‌های Debugging و ویرال استودیو همانند کدهای مدیریت شده‌ی دات نت جهت برنامه نویسی مبتنی بر WinRT در اختیار برنامه نویسی‌ها خواهد بود (فرمت ارائه شده، [ECMA 335 metadata format](#) می‌باشد). همچنین اشیاء COM متعلق به WinRT به خوبی توسط GC (آشغال جمع کن) دات نت جهت مدیریت بهتر حافظه، تحت نظر می‌باشند. بنابراین از دیدگاه یک برنامه نویسی دات نت، کل WinRT به صورت managed assemblies مشاهده می‌شود، اینترفیس‌های آن همان اینترفیس‌های دات نتی خواهند بود و کلاس‌های آن نیز به همین ترتیب. این مشکلی بود/هست که با Win32 API در دات نت وجود دارد و دسترسی به آن به این سادگی و یکپارچگی میسر نیست (هر چند تا الان کل اینترفیس آن جهت استفاده در دات نت نیز ترجمه شده است). در اینجا شما ارجاعاتی را به فایل‌هایی با پسوند winmd یا windows metadata، به پروژه‌ی دات نتی خود اضافه می‌کنید و سپس CLR قادر خواهد بود تا کلیه اطلاعات لازم جهت کار با WinRT را از آن‌ها استخراج کند (این فایل‌ها در پوشه C:\Program Files (x86)\Windows Kits\8.0\Windows Metadata و C:\Windows\system32\winmetadat و ویندوز 8 قابل مشاهده و دسترسی هستند).

تفاوت‌های مهم امکانات نمایشی WinRT با Win32 API کدامند؟

تفاوت مهم WinRT با Win32 API از دیدگاه برنامه نویسی‌ها، امکان دسترسی بیشتر به آن از طریق زبان‌های مختلف می‌باشد. WinRT همانند Win32 API توسط CPP، دات نت و سایر روش‌های مرسوم دیگر قابل دسترسی و توسعه است. اما اینبار WinRT برخلاف Win32 API، از طریق HTML و جاوا اسکریپت هم قابل توسعه است. در این حالت کدهای شما توسط Chakra JavaScript engine که از اینترنت اکسپلورر 9 به بعد ارائه شده، اجرا خواهد شد. بنابراین «برفراز» WinRT دو لایه نمایشی (presentation layer) قابل طراحی و دسترسی است. XAML و زبان‌های متداول برنامه نویسی موجود مانند سی شارپ و وی بی دات نت و غیره. همچنین HTML/CSS هم مجال ابراز وجود یافته است. البته XAML تنها لایه نمایشی کلیه زبان‌های قدیمی موجود مانند سی شارپ، وی بی دات نت، CPP و غیره خواهد بود. به همین جهت [Expression Blend جدید](#) نیز از HTML 5 پشتیبانی می‌کند. همچنین در WinRT، قسمت‌های GDI و Message loop متداول Win32 API حذف شده است و از DirectX استفاده می‌کند. برای نمونه کدهای XAML شما توسط DirectX رندر می‌شود. البته این مطلب جدیدی نیست و از زمان ارائه WPF شاهد این مساله بوده‌ایم.

وضعیت توسعه پذیری WinRT چگونه است؟

علاوه بر این‌ها، برخلاف Win32 API، اینبار WinRT قابل توسعه است و [Extensions SDK](#) برای آن ارائه شده است.

آیا WinRT شاهد تغییرات امنیتی خاصی هم بوده است؟

نکته‌ی مهمی که در طراحی WinRT به آن توجه شده است، امنیت می‌باشد. برنامه‌های WinRT شبیه به برنامه‌های سیلورلایت در یک [Sandbox](#) اجرا می‌شوند. به این معنا که جهت ذخیره سازی اطلاعات خود از یک isolated storage استفاده می‌کنند. برای کار با file system نیاز به تأیید کاربر دارند و خلاصه دیگر به سادگی نمی‌توان از مرزهای این نوع برنامه‌ها رد شد و سیستم عاملی را root کرد. برای نمونه برنامه نویسی‌های دات نت دسترسی به فضای نام System.IO.FileStream را دیگر نخواهند داشت و تنها قسمتی از دات نت «برفراز» WinRT و [مدل امنیتی جدید آن](#) معنا پیدا می‌کند. همچنین برفراز این API جدید، تولید مثلاً Device drivers هم دیگر معنا پیدا نمی‌کند. این محدودیت‌های امنیتی برای برنامه نویسی‌های native هم وجود دارد و تفاوتی نمی‌کند. کلاً برنامه‌های جدید Metro-style در یک قرنطینه‌ی کامل امنیتی اجرا می‌شوند. برای مثال اگر برنامه‌ای نیاز به دسترسی به یک WebCam را داشته باشد، همانند برنامه‌های سیلورلایت ابتدا باید کاربر تأیید کرده و سپس برنامه مجوز امنیتی کار با مثلاً یک WebCam را خواهد یافت. همچنین تمام برنامه‌های جدید Metro-style باید جهت ارائه در فروشگاه جدید ویندوز 8، دارای امضای دیجیتالی معتبر نیز باشند.

آیا جهت توسعه‌ی برنامه‌های چندرسمانی و غیرهمزمان تمهیدات خاصی در WinRT پیش‌بینی شده است؟

در طراحی جدید WinRT به اعمال [asynchronous](#) به شدت توجه شده است. هر عملی که بیش از 50 میلی ثانیه طول بکشد به صورت خودکار تبدیل به یک عمل [asynchronous](#) خواهد شد تا برنامه‌ها مرتباً در حین اجرای اعمال زمانبر هنگ نکرده و ترد اصلی برنامه را بلاک نکنند. علاوه بر این‌ها WinRT از طریق IAsyncOperation interface خود، امکان استفاده از واژه‌های جدید کلیدی async/await سی شارپ 5 را نیز مهیا می‌سازد.

آیا WinRT آمده است تا جایگزینی برای دات نت و سیلورلایت و امثال آن باشد؟

خیر. WinRT نگارش دوم Win32 API است با هدف توسعه پذیری، استفاده از دایرکت ایکس و فناوری‌های جدید که عموماً از شتاب دهنده‌های سخت افزاری هم بهره‌مند هستند بجای GDI سابق، استفاده ساده‌تر در زبان‌های دیگر به کمک فایل‌های استاندارد Windows Meta data آن می‌باشد. همچنین این API جدید دسترسی به امکانات ویندوز را هم توسط HTML و جاوا اسکریپت، علاوه بر امکانات مهمی سابق میسر ساخته است. هم اکنون [نگارش 4 و نیم دات نت](#) در ویندوز 8 ارائه شده است و توسط هر دو سیستم سابق و جدید قابل استفاده می‌باشد. البته باید در نظر داشت که جهت استفاده از WinRT به دلایل محدودیت‌های امنیتی اعمال شده به آن و همچنین استفاده از XAML به تنها عنوان لایه نمایشی سیستم‌های متداول غیر HTML ایی، دات نت فریم ورک به امکانات و کلاس‌های کمتری نسبت به حالت متداول کار با آن، دسترسی دارد (جهت درک بهتر این محدودیت‌ها می‌توان به طراحی سیلورلایت مراجعه کرد). این را هم باید اضافه کرد که ویندوز 8 توانایی اجرای هر دو نوع برنامه‌های سبک جدید مترو و متداول دسکتاپ قدیمی را دارا است.

جهت آشنایی بیشتر با WinRT می‌توان به مجموعه‌ای از ویدیوهای مرتبط آن مراجعه کرد:

<http://channel9.msdn.com/Events/BUILD/BUILD2011?t=windows%2Bruntime>

نظرات خوانندگان

نویسنده: afsharm

تاریخ: ۱۱:۱۸:۴۶ ۱۳۹۰/۰۶/۲۹

آیا واقعاً WinRT جایگزینی برای WPF نخواهد بود؟

نویسنده: وحید نصیری

تاریخ: ۱۲:۰۶:۲۸ ۱۳۹۰/۰۶/۲۹

خیر. به WinRT یا Windows run time جدید به چشم Silverlight run time نگاه کنید. لایه‌ای است بالاتر از Win32 API که برفراز آن می‌شود برنامه توسعه داد (WPF فقط یک روش طراحی رابط کاربری جدید است، run time نیست). WinRT شبیه به همان مدل امنیتی بسته سیلورلایت را به ارث برده. همان لایه نمایش XAML را به ارث برده. با این تفاوت که این run time جدید آن علاوه بر XAML، امکان کار با HTML5 و جاوا اسکریپت را هم می‌دهد و محدود به XAML نیست. برخلاف سیلورلایت، این run time فقط محدود به دات نت هم نیست و با CPP و Native کد هم می‌شود با آن کار کرد. به این نتیجه رسیدند که طراحی خوبی انجام دادن، خوب چرا فقط دات نت استفاده کند؟

ضمناً این محدودیت‌های امنیتی ذکر شده برای آن قطعا برای خیلی از برنامه نویس‌ها خوشایند نخواهد بود. برای دسترسی بیشتر، مجبور خواهند شد به سیستم‌های دسکتاپ سابق رجوع کنند.

نویسنده: وحید نصیری

تاریخ: ۱۶:۱۵:۲۴ ۱۳۹۰/۰۶/۲۹

علاوه بر این‌ها امکانات فعلی WinRT کمتر از نمونه‌ی موجود در دات نت است. کتابخانه‌های XAML آن کلاً با CPP بازنویسی شده و متکی بر دات نت نیست. WinRT XAML تنها قسمتی از XAML در دسترس دات نت را ارائه می‌دهد مثلاً DataTriggers و غیره آن فعلاً پیاده سازی نشده. همچنین برفراز WinRT شما تنها به قسمتی از کل دات نت فریم ورک دسترسی دارید که به آن اشاره شد (همان sandbox معروف). خلاصه توانایی‌های XAML آن به هیچ عنوان جایگزین کامل WPF دات نت نمی‌تواند باشد.

نویسنده: وحید نصیری

تاریخ: ۱۶:۲۷:۵۶ ۱۳۹۰/۰۶/۲۹

شاید عده‌ای خوشحال شده باشند که مثلاً این XAML الان Native شده، چقدر خوب! جهت اطلاع مقاله زیر را مطالعه کنید تا به قدر و منزلت دات نت بیشتر پی ببرید!

[Real Native WinRT Development](#)

نویسنده: Smart_boy592

تاریخ: ۱۶:۳۰:۱۱ ۱۳۹۰/۰۶/۲۹

سلام {{}}همچنین تمام برنامه‌های جدید Metro-style باید جهت ارائه در فروشگاه جدید ویندوز 8، دارای امضای دیجیتال معتبر نیز باشند.}}

با این وجود تکلیف برنامه نویسی‌های ایرانی چه می‌شود؟ ماها خودمون فعلاً سر ویندوز و VS اورجینال مورد داریم D:

نویسنده: وحید نصیری

تاریخ: ۱۶:۳۲:۰۹ ۱۳۹۰/۰۶/۲۹

در مورد این HTML که در بالا در مورد آن صحبت شد لازم است بیشتر توضیح داده شود: برخلاف تصور عموم برنامه‌های HTML/CSS/JavaScript ویندوز 8 منحصرأ برای WinRT تهیه می‌شوند و ... و ... قابل انتقال به سایر سکوها کاری «نیستند». این‌ها از API مخصوص WinRT استفاده می‌کنند تا معنا پیدا کنند. بنابراین این مورد اصلاً web programming متداول نیست. به آن باید به چشم یک standalone Windows 8 app نگاه کرد.

نویسنده: وحید نصیری
تاریخ: ۱۶:۳۶:۰۳ ۱۳۹۰/۰۶/۲۹

- جهت تهیه سرتیفیکت دیجیتال معتبر هستند شرکت‌هایی در ایران که این خریدها رو انجام بدن و به صورت ریسر عمل کنند.
- نمونه‌اش تهیه سرتیفیکت‌های SSL برای وب سرورها است.
- فعلا مشکلی که معلوم نیست این است که اصلا درب این فروشگاه به روی ایرانی‌ها باز خواهد بود یا خیر. این رو نمی‌دونم.

نویسنده: وحید نصیری
تاریخ: ۱۶:۵۴:۱۹ ۱۳۹۰/۰۶/۲۹

اگر علاقمندید که بدانید چه قسمت‌هایی از دات نت فریم ورک برفراز WinRT و Sandbox آن در دسترس هستند به این مقاله مراجعه کنید:

[Metro .NET Framework Profile](#)

نویسنده: amin
تاریخ: ۱۸:۲۲:۲۳ ۱۳۹۰/۰۶/۲۹

جناب وحید نصیری قبل از هر چی از مطلب خوبتون تشکر میکنم سایت یا بهتره بگم وبلاگ خیلی خوبی دارید من خیلی چیزها توش یاد گرفتم.

مهندس بنده تقریبا تازه کارم و 1 سالی هست که دارم با C# کار میکنم ولی امروز که این مطلب رو خوندم و بخصوص مطلب سایت نارنجی

<http://tinyurl.com/3oufmo5> خیلی نگران شدم وضعیت زبان C sharp چی میشه دوست ندارم روی چیزی کار کنم که 1-2 سال دیگه کاربردی نداشته باشه یا بگن این قدیمی شده و مثل الان زبان VB بشه؟ به نظر شما قراره چیز دیگه ای جایگزین بشه؟ میشه یکم راهنمایی کنید.

یعنی مایکروسافت میخواد دات نت و C sharp رو کم کم کنار بزاره و یه چیز جدید جایگزین کنه؟

نویسنده: وحید نصیری
تاریخ: ۱۸:۳۳:۰۸ ۱۳۹۰/۰۶/۲۹

حدسم این است که مطالب فوق را مطالعه نکرده‌اید. لطفا تمام توضیحات فوق را مطالعه کنید + نسخه‌ی جدید سی شارپ به زودی ارائه میشه. همچنین ویندوز 8 دارای دات نت 4 و نیم سرخود است. اطلاعات بیشتر رو می‌تونید از زبان خود خالق سی‌شارپ بشنوید:

[Future directions for C# and Visual Basic](#)

نویسنده: وحید نصیری
تاریخ: ۲۲:۵۷:۳۶ ۱۳۹۰/۰۶/۲۹

[آدرس انجمن‌های رسمی توسعه برنامه‌های جدید مترو](#)

نویسنده: وحید نصیری
تاریخ: ۲۲:۵۸:۴۴ ۱۳۹۰/۰۶/۲۹

[آدرس دریافت برنامه‌های آماده سبک مترو](#)

نویسنده: وحید نصیری
تاریخ: ۱۶:۲۸:۴۵ ۱۳۹۰/۰۶/۳۰

سؤال الان مطرح هست که چه نوع برنامه‌هایی بهتر است به سبک مترو تهیه شوند و واقعا هدفگیری اصلی این روش چیست؟
مطلبی رو در این مورد از زبان یکی از مدیران شرکت معروف Telerik در اینجا مطالعه کنید:

[There is a need for only five Metro style apps in the world](#)

نویسنده: مهمان
تاریخ: ۱۷:۳۰:۲۹ ۱۳۹۰/۰۶/۳۰

با تشکر از شما استاد گرامی
به نظر شما با توجه به سیستم عامل جدید بهتره روی کدام مقوله متمرکز بشم

1- سیلورلایت

wpf 2-

asp.NET 3-

++C

با تشکر

نویسنده: وحید نصیری
تاریخ: ۲۱:۵۵:۴۴ ۱۳۹۰/۰۶/۳۰

- بحث وب که سر جای خودش همانند سابق هست و هیچ فرقی نمی‌کند. برنامه‌های ASP.NET روی سرور اجرا می‌شوند و عموماً روی سرور بجز یک سری سرویس‌های ویندوز NT، هیچ نرم افزار دیگری نصب نخواهد شد. مثلاً IIS یا مثلاً SQL Server و در همین حد. حتی عموماً سرورها حتی مونیتور هم ندارند و با ریموت دسکتاپ یک سری کارهای مدیریتی آن‌ها را انجام می‌دهند و این کارها هم طوری نیست که هر روز تغییر کند. یکبار سرور تنظیم می‌شود که حداقل یکسال یا بیشتر کار کند. این مورد اصلاً تغییری نخواهد داشت. بحث سمت سرور است. بنابراین سرمایه گذاری روی ASP.NET خوب است و شامل این بحث ویندوز 8 یا ویندوزهای بعدی نمی‌شود؛ چون این‌ها (WinRT) سمت کاربر محسوب می‌شوند.

- از این جهت که رابط‌های کاربری مبتنی بر WinRT، یا بر پایه XAML است یا HTML/CSS، یادگیری WPF و یا سیلورلایت (که قسمتی از WPF را به ارث برده) مفید خواهند بود؛ از این لحاظ که پایه رابط کاربری هر دوی این‌ها هم XAML است و اساساً طراحی XAML اینجا به WinRT منتقل شده.

کلاً برای برنامه نویسی‌های دات نت WinRT مثل یک سری اسمبلی جدید است که اضافه شده و یک سری اسمبلی از آن‌ها گرفته شده. هیچ تفاوت دیگری از لحاظ اصول برنامه نویسی نمی‌کند. یک سری فضای نام جدید و کلاس جدید دارید. یک سری از کلاس‌های پیشین به دلیل محدودیت‌های امنیتی، دیگر در WinRT قابل استفاده نیست. مثلاً همینطوری دیگه نمی‌تونید هر جایی فایل جدید درست کنید، یک سری آداب و اصول خاص خودش را دارد.

ضمناً این رو هم در نظر داشته باشید که WinRT یک سیستم همه منظوره نیست و ... بین خودمان باشد بیشتر در سطح دسکتاپ برای کارهای شیک و چشم نواز و برنامه‌های فانتزی طراحی شده. اصل کارهای برنامه‌های تجاری باز هم بر اساس همان سیستم‌های وب و یا دسکتاپ سابق خواهد بود.

- یادگیری سی++ همیشه مفید است. حتی در کره مریخ هم تاجایی که اطلاع دارم (!) یک کامپایلر سی++ وجود دارد و می‌شود با آن برنامه‌ی Hello world را کامپایل کرد. اگر باور ندارید از این لینوکسی‌ها بپرسید!

نویسنده: وحید نصیری
تاریخ: ۱۲:۰۸:۲۷ ۱۳۹۰/۰۶/۳۱

معنای WinRT برای برنامه نویسی‌های دات نت چیست؟

WinRT، دات نت نیست. برای درک این موضوع باید CLR و دات نت فریم ورک را از هم جدا کرد. برنامه‌های دات نت نوشته شده برای WinRT برفراز CLR اجرا می‌شوند اما از دات نت فریم ورک استفاده نمی‌کنند. بجای آن از توانمندی‌های مشابه موجود در WinRT، در پشت صحنه استفاده خواهند کرد.

در اینجا برنامه‌های XAML + C++ بدون دخالت CLR و مستقیماً برفراز WinRT کار خواهند کرد. برنامه‌های سبک مترو HTML/CSS/JavaScript هم به همین صورت متکی به CLR نیستند و مستقیماً برفراز WinRT اجرا می‌شوند.

WinRT فقط قادر است برنامه‌های سبک مترو و ویندوز 8 را هاست کند. اگر نیاز دارید سیستم عامل‌های قدیمی را پشتیبانی کنید یا

اینکه اصلاً کارتان ساخت برنامه‌های سمت کاربر و دسکتاپ نیست، اصلاً این تغییرات به کار شما مرتبط نخواهند شد.

الگوی اصلی تعاملی برنامه‌های سبک مترو با تمرکز بر برنامه‌های لمسی (touch focused) و مبتنی بر محتوا (content-oriented) است. به این معنا که تمام برنامه‌های تجاری موجود که دارای 10 ها و صدها صفحه‌ی ورود اطلاعات هستند، اصلاً برای این نوع سبک ارائه محتوا طراحی نشده‌اند و نخواهند شد. کاربردهای مهم این سبک، استفاده از آن در برنامه‌های مخصوص تمام صفحه tablets است یا حداکثر در حد داشبوردهای ارائه خلاصه گزارشات یک برنامه می‌توانند اهمیت داشته باشند. بنابراین اگر کارتان در حیطه‌ی ساخت برنامه‌های مخصوص tablets و برنامه‌های لمسی قرار نمی‌گیرد، کماکان به ساخت برنامه‌های دسکتاپ نوشته شده با WPF/WinForms می‌توانید مشغول باشید.

انتقال قسمت عمده‌ای از برنامه‌های موجود WPF و یا Silverlight مبتنی بر الگوهایی مانند MVVM و امثال آن با توجه به عدم گره خوردگی آن‌ها به لایه نمایشی، به WinRT میسر است.

در سمت سرور، تمرکز اصلی هنوز همان دات نت فریم ورک است. قرار نیست ASP.NET یا WCF و سایر مؤلفه‌های اصلی دات نت به WinRT منتقل شوند. حتی اگر WinRT به سرورهای بعدی هم راه پیدا کند در حد همان لایه نمایشی مترو است و تاثیری بر روی سرویس‌های NT با دسترسی بالای ویندوز، نخواهد داشت. مثلاً قرار نیست SQL Server را با WinRT پیاده سازی کنند.

نویسنده: Rz_mohammad
تاریخ: ۱۳۹۰/۰۶/۳۱ ۱۲:۱۷:۵۴

ممنون از مطلب زیبا و خوبتون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۶/۳۱ ۲۳:۴۶:۵۶

پرسش و پاسخ‌های WinRT در Stack overflow: [^] و همچنین فید آن هم مهیا است: [^]

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۷/۰۳ ۲۰:۳۷:۵۰

تصویری از این [Windows Metadata](#): [^]

نویسنده: A.Karimi
تاریخ: ۱۳۹۰/۰۷/۰۸ ۱۸:۳۵:۱۴

Miguel de Icaza در وبلاگش در مورد Async API اینچنین گفته :

With WinRT, Microsoft has followed a simple rule: if an API is expected to take more than 50 milliseconds to run, the API is asynchronous. (<http://tirania.org/blog/archive/2011/Sep-15.html>) ظاهراً عملیات به طور خودکار تبدیل به عملیات Asynchronous نمی‌شوند بلکه اگر یک API قرار باشد بیشتر از 50 میلی ثانیه طول بکشد باید انتظار داشته باشیم که فقط امکان فراخوانی به شکل Asynchronous را داشته باشد.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۷/۰۸ ۱۹:۴۰:۴۷

در سیلور لایت همه چیز Asynchronous است. از فراخوانی یک وب سرویس تا تبادلات شبکه تا هر نوع API زمانبر دیگر. به عمد تمام متدهای همزمان رو که در دات نت وجود دارند حذف کردند. این خوبه. از ابتدا شخص عادت می‌کنه تا فکرش به این سمت سوق پیدا کنه. سی شارپ 5 هم این نوع فراخوانی‌ها رو ساده تر و طبیعی تر می‌کنه.

نویسنده: A.Karimi

تاریخ: ۲۲:۲۰:۵۷ ۱۳۹۰/۰۷/۰۸

بله. البته منظور بنده نه Silverlight بلکه WinRT بود. ظاهراً در WinRT هنوز API‌های بسیار سریعی وجود دارد که Synchronous باشد!

نویسنده: وحید نصیری
تاریخ: ۲۳:۱۶:۴۱ ۱۳۹۰/۰۷/۰۸

- البته «سرعت» در اینجا بیشتر بحث سرعت اتصال به شبکه و مواردی مانند آن است (کار با فایل سیستم، بانک اطلاعاتی، وب کم و ...) و آنچنان ارتباطی به کلاینت مورد استفاده ندارد. یعنی عامل‌های تعیین کننده اصلی سرعت، خارج از مرزهای سیستم قرار دارند.

- هدفم از پاسخ فوق، اشاره به پیش زمینه‌ی ذهنی بود که هم اکنون سبب حذف امکان وجود فراخوانی‌های همزمان شده است؛ مانند: [انتخاب فایل](#)، [کار با وب کم](#). این نوع کدها برای برنامه نویسی‌های سیلورلایت بسیار آشنا هستند.

نویسنده: a. hosnoddinov
تاریخ: ۱۵:۳۰:۱۰ ۱۳۹۰/۰۷/۱۱

جناب نصیری از مطلب کامل، مختصر و مفید شما ممنونم. منم نظر خودمون رو اینجا عنوان کنم... بسیاری از مسائل از تحلیل و برداشت غلط نشئت می گیرند... تصور اینکه میکروسافت بخواهد دات نت فریم ورک و یا زبان های دات نت رو جمع کنه در حالیکه باید ده ها سال ازین تکنولوژی پشتیبانی ارائه کنه خیلی سخت و بعید به نظر می رسد. WinRT همانگونه که عنوان شد فقط بستر طراحی اپلیکیشن های کلاینت مبتنی بر واسط کاربری مترو بوده و هیچ ارتباطی با دات نت فریم ورک ندارد. در بحث سمت سرور دات نت فریم ورک و مباحث مرتبط با آن نه تنها تضعیف نشده بلکه حرف اصلی را می زنند و پررنگتر از قبل نیز خواهند بود. این همه ورژن جدید در کنفرانس build ارائه نشد که با ویندوز 8 جمع شه... مقاله ای در خصوص ویژگی های جدید سی شارپ رو اخیرا منتشر کردم که مطالعه ی آن نیز می تواند برای درک مطالب جدید مفید باشد.

<http://www.persiadevelopers.com/articles/cs5-after-build.aspx>

نویسنده: وحید نصیری
تاریخ: ۱۲:۲۰:۰۵ ۱۳۹۰/۰۷/۱۲

[راهنمای انتقال برنامه‌های دات نت به سبک مترو](#)

نویسنده: وحید نصیری
تاریخ: ۲۲:۴۳:۳۰ ۱۳۹۰/۰۷/۱۵

فلوچارت جالبی اینجا جهت انتخاب فناوری‌های مناسب، برای تهیه برنامه‌های مختلف ارائه شده:
[چه زمانی باید برنامه‌های خود را بر اساس سبک مترو ویندوز 8 تهیه کرد؟](#)