عنوان: **خلاصه‌ای کوتاه در مورد WinRT**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۶/۲۹ ۱۰:۴۴:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: WinRT

WinRT چیست؟

مایکروسافت جهت سهولت تولید برنامه‌های جدید Metro-style ، لمسی (touch-centric) و tablets ویندوز 8 ، اقدام به بازنویسی مجدد Windows-API کرده و نام آن‌را 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 اصلا دات نتی در کار نبود)، جهت سازگاری با دات نت طراحی شده است. این طراحی جدید ILDasm metadata را در اختیار برنامه نویس‌های دات نت قرار می‌دهد و به این ترتیب IntelliSense و قابلیت‌های Debugging ویژوال استودیو همانند کدهای مدیریت شده‌ی دات نت جهت برنامه نویسی مبتنی بر WinRT در اختیار برنامه نویس‌ها 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 نویس 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 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 نویس‌ها، 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 تائید کاربر دارند و خلاصه دیگر به سادگی نمی‌توان از مرزهای این نوع برنامه‌ها رد شد و سیستم عاملی را root کرد. برای نمونه برنامه نویس‌های 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 مثلا Device drivers هم دیگر معنا پیدا نمی‌کند. این محدودیت‌های امنیتی برای برنامه نویس‌های native هم وجود دارد و تفاوتی نمی‌کند. کلا 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 تائید کرده و سپس برنامه مجوز امنیتی کار با مثلا یک WebCam را خواهد یافت. همچنین تمام برنامه‌های جدید Metro-style باید جهت ارائه در فروشگاه جدید ویندوز 8، دارای امضای دیجیتال معتبر نیز باشند.

آیا جهت توسعه‌ی برنامه‌های چندریسمانی 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 مرتبا 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 عموما 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 مهیای 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: ۱۳۹۰/۰۶/۲۹ ۱۱:۱۸:۴۶

آیا واقعاً WinRT جایگزینی برای WPF نخواهد بود؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۶/۲۹ ۱۲:۰۶:۲۸

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؟
ضمنا 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: ۱۳۹۰/۰۶/۲۹ ۱۶:۱۵:۲۴

علاوه بر این‌ها امکانات فعلی WinRT کمتر از نمونه‌ی موجود در دات نت است. کتابخانه‌های XAML آن کلا با CPP بازنویسی شده و متکی بر دات نت نیست. WinRT XAML تنها قسمتی از XAML در دسترس دات نت را ارائه می‌دهد مثلا DataTriggers و غیره آن فعلا 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: ۱۳۹۰/۰۶/۲۹ ۱۶:۲۷:۵۶

شاید عده‌ای خوشحال شده باشند که مثلا 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: ۱۳۹۰/۰۶/۲۹ ۱۶:۳۰:۱۱

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 نویس های ایرانی چه می شود ؟ ماها خودمون فعلا سر ویندوز و VS اورجینال مورد داریم :D

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۶/۲۹ ۱۶:۳۲:۰۹

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 منحصرا برای WinRT تهیه می‌شوند و ... و ... قابل انتقال به سایر سکوهای کاری «نیستند». این‌ها از API مخصوص WinRT استفاده می‌کنند تا معنا پیدا کنند. بنابراین این مورد اصلا web programming متداول نیست. به آن باید به چشم یک Windows 8 app standalone نگاه کرد.

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

- جهت تهیه سرتیفیکت دیجیتال معتبر هستند شرکت‌هایی در ایران که این خریدها رو انجام بدن و به صورت ریسلر عمل کنند. نمونه‌اش تهیه سرتیفکیت‌های 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: ۱۳۹۰/۰۶/۲۹ ۱۶:۵۴:۱۹

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: ۱۳۹۰/۰۶/۲۹ ۱۸:۲۲:۲۳

جناب وحید نصیری قبل از هر چی از مطلب خوبتون تشکر میکنم سایت یا بهتره بگم وبلاگ خیلی خوبی دارید من خیلی چیزا 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/3oufmo5خیلی 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: ۱۳۹۰/۰۶/۲۹ ۱۸:۳۳:۰۸

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: ۱۳۹۰/۰۶/۲۹ ۲۲:۵۷:۳۶

آدرس انجمن‌های رسمی توسعه برنامه‌های جدید مترو

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

آدرس دریافت برنامه‌های آماده سبک مترو

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

سؤالی 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: ۱۳۹۰/۰۶/۳۰ ۱۷:۳۰:۲۹

با تشکر از شما استاد گرامی
به نظر شما با توجه به سیستم عامل جدید بهتره روی کدام مقوله متمرکز بشم
1- سیلورلایت
2- wpf
3- asp.NET
C++

با تشکر

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

- بحث وب که سرجای خودش همانند سابق هست و هیچ فرقی نمی‌کند. برنامه‌های ASP.NET روی سرور اجرا می‌شوند و عموما روی سرور بجز یک سری سرویس‌های ویندوز NT ، هیچ نرم افزار دیگری نصب نخواهد شد. مثلا IIS یا مثلا SQL Server و در همین حد. حتی عموما 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 اصلا 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 اساسا طراحی XAML از اینجا به WinRT منتقل شده.

کلا برای برنامه نویس‌های 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. مثلا همینطوری دیگه نمی‌تونید هر جایی فایل جدید درست کنید، یک سری آداب و اصول خاص خودش را دارد.
ضمنا 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: ۱۳۹۰/۰۶/۳۱ ۱۲:۰۸:۲۷

معنای WinRT برای برنامه نویس‌های 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 مستقیما برفراز WinRT کار خواهند کرد. برنامه‌های سبک مترو HTML/CSS/JavaScript هم به همین صورت متکی به CLR نیستند و مستقیما برفراز WinRT اجرا می‌شوند.

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

اینکه اصلا کارتان ساخت برنامه‌های سمت کاربر و دسکتاپ نیست، اصلا 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، اصلا 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 داشبردهای 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 موجودWPF 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. مثلا 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 عملیاتها 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 سیلورلایت 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 ساده‌تر و طبیعی‌تر می‌کنه.

نویسنده: A.Karimi

تاریخ: ۱۳۹۰/۰۷/۰۸ ۲۲:۲۰:۵۷

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: ۱۳۹۰/۰۷/۰۸ ۲۳:۱۶:۴۱

- 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 نویس‌های سیلورلایت بسیار آشنا هستند.

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

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 مایکروسافت بخواهد دات نت فریم ورک و یا زبان های دات نتی 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 مربتط با آن نه تنها تضعیف نشده بلکه حرف اصلی را می زنند و پررنگتر از قبل نیز خواهند بود. این همه ورژن جدید در کنفرانیس 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: ۱۳۹۰/۰۷/۱۲ ۱۲:۲۰:۰۵

راهنمای انتقال برنامه‌های دات نت به سبک مترو

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

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؟

عنوان: **هیتلر و WinRT**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۷/۰۶ ۰۱:۲۲:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: WinRT

یک سری ویدیو در یوتیوب هست که پایه اصلی آن‌ها قسمتی از بحث هیتلر با فرماندهان SS است. حالا اینجا افراد مختلف اومدن برای این یک تک ویدیو، زیرنویس‌های مختلفی تهیه و آپلود کرده‌اند. یکی از این‌ها، به همین بحث داغ WinRT مرتبط است. این زیر نویس رو به فارسی ترجمه کردم که به همراه اصل ویدیو از آدرس‌های زیر قابل دریافت هستند:

ویدیوی اصلی
+ زیرنویس فارسی

برای دیدن بدون دردسر زیر نویس تهیه شده هم می‌شود از برنامه‌ی عالی و رایگان KMPlayer استفاده کرد.

نمونه‌های مشابه دیگر در یوتیوب:
Hitler teaches us about Agile Development
Hitler, Waterfall and Scrum
Hitler in software project disaster
Hitler at the Daily Scrum
Hitler's nightly build fails
Hitler does unit testing

نظرات خوانندگان

نویسنده: Ameer Taghavi
تاریخ: ۱۳۹۰/۰۷/۰۶ ۱۳:۵۳:۵۱

خیلی با مزه و زیبا بود. :دی

نویسنده: A.Karimi
تاریخ: ۱۳۹۰/۰۷/۰۸ ۱۸:۲۹:۳۴

بسیار جالب بود

نویسنده: Ramin Alirezaee
تاریخ: ۱۳۹۰/۰۷/۰۸ ۲۳:۴۲:۵۱

آقا من با این ویدئو دچار افسردگی شدم. خیلی دارم برای سیلورلایت وقت میزارم. هر چند من که فکر نمیکنم اینها بتونن جای سیلورلایت رو پر کنن

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۷/۰۸ ۲۳:۵۹:۴۷

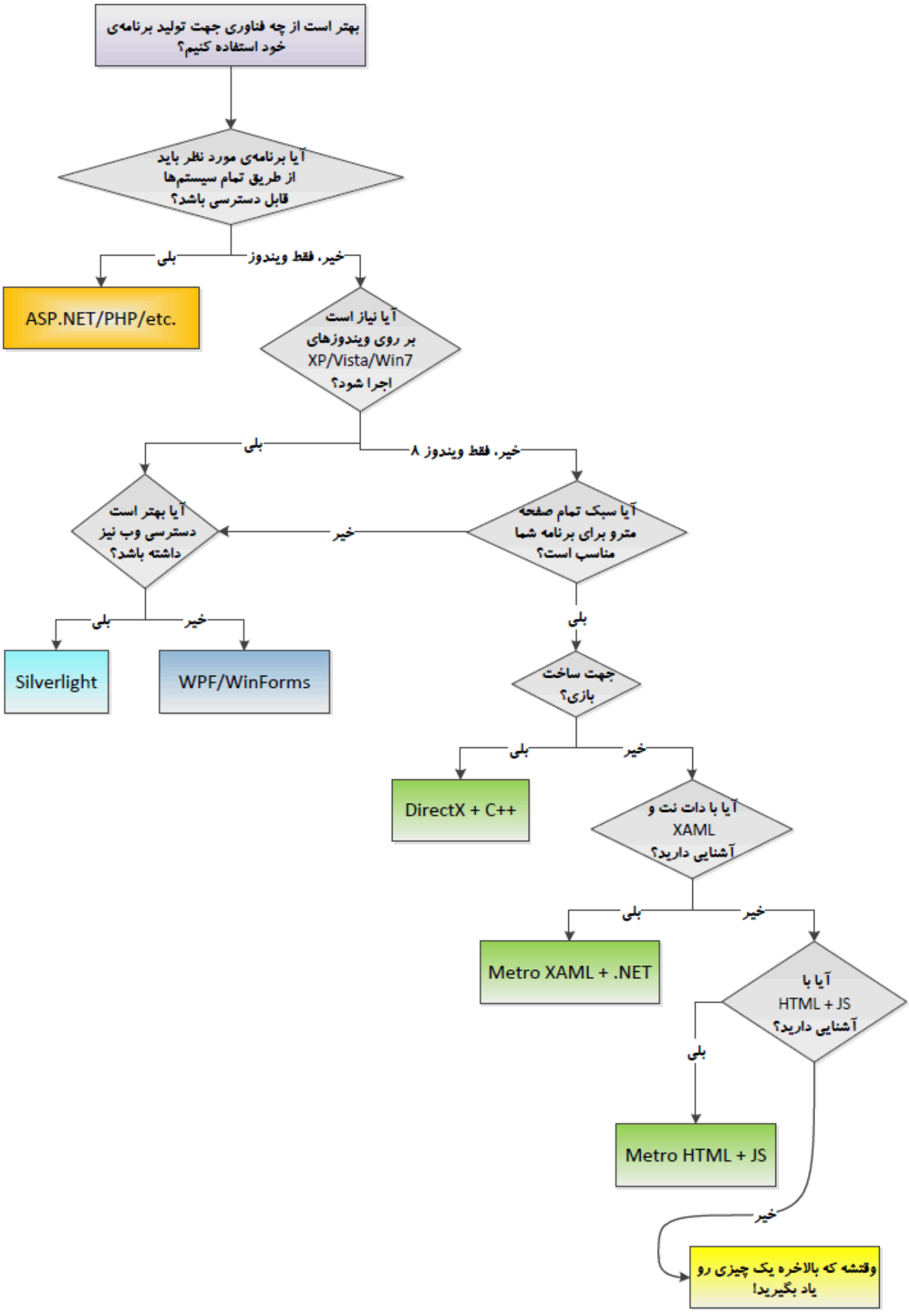
Win8, Metro Apps and Silverlight

نویسنده: Mahdi Z K
تاریخ: ۱۳۹۰/۰۷/۱۱ ۱۳:۴۵:۱۸

سلام، با وبلاگ شما از طریق قسمت پرسش و پاسخ ضمیمه کلیک روزنامه جام جم(10 مهر) آشنا شدم. به دلیل علاقه زیاد به نرم افزارهای ایرانی وب شما رو لینک می کنم، با تشکر.

عنوان: **راهنمای انتخاب فناوری‌های مختلف**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۷/۲۹ ۱۱:۲۶:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: WinRT

اگر مطالب مرتبط با WinRT را دنبال کرده باشید، شرکت Telerik فلوچارتی را به عنوان راهنمای انتخاب فناوری‌های مختلف جهت توسعه برنامه‌های ویندوز با جهت گیری دات نت، منتشر کرده‌است که برگردان فارسی آن به صورت زیر است:


بهتر است از چه فناوری جهت تولید برنامه‌ی خود استفاده کنیم؟
آیا برنامه‌ی مورد نظر باید از طریق تمام سیستم‌ها قابل دسترسی باشد؟
بلی
خیر، فقط ویندوز
ASP.NET/PHP/etc.
آیا نیاز است بر روی ویندوزهای XP/Vista/Win7 اجرا شود؟
بلی
خیر، فقط ویندوز ۸
آیا بهتر است دسترسی وب نیز داشته باشد؟
خیر
آیا سبک تمام صفحه مترو برای برنامه شما مناسب است؟
بلی
خیر
Silverlight
WPF/WinForms
بلی
جهت ساخت بازی؟
بلی
خیر
DirectX + C++
آیا با دات نت و XAML آشنایی دارید؟
بلی
خیر
Metro XAML + .NET
آیا با HTML + JS آشنایی دارید؟
بلی
خیر
Metro HTML + JS
وقتشه که بالاخره یک چیزی رو یاد بگیرید!

همین تصویر با فرمت‌های PDF و Visio

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۷/۲۹ ۱۲:۰۳:۵۳

البته اگر بیشتر بحث سازمان‌ها مطرح باشد:
چرا در سازمان‌ها برنامه‌های وب جایگزین برنامه‌های دسکتاپ شده‌اند (یا می‌شوند)؟

نویسنده: mohammadsaheb
تاریخ: ۱۳۹۰/۰۷/۲۹ ۱۶:۱۴:۲۲

ممنون
یه سوال Silverlight نمیتونه تو قسمت بلی "دسترسی از طریق تمام سیستم ها" قرار بگیره؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۷/۲۹ ۱۶:۵۰:۳۴

چرا اتفاقا، چون خود مایکروسافت نسخه‌ی Mac آن‌را منتشر کرده و پشتیبانی می‌کند. لینوکسی‌ها هم تحت پروژه Moonlight به آن دسترسی دارند. فقط نسبت به HTML خالص دسترسی موبایل آن شاید کمتر باشد.
البته چون سیلورلایت قابلیت اجرای خارج از مرورگر را هم دارد، در رده‌ی برنامه‌های معمولی ویندوز هم می‌تواند قرار گیرد.

نویسنده: Hossein Moradinia
تاریخ: ۱۳۹۰/۰۷/۲۹ ۱۹:۴۱:۲۶

تشکر
زیبا بود

عنوان: **زیر نویس فارسی ویدیوهای ساخت برنامه‌های مترو توسط سی شارپ و XAML - قسمت اول**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۱۰/۲۳ ۱۰:۲۲:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: WinRT

سایت pluralsight ویدیوهای آموزشی بسیار با کیفیتی را در مورد مباحث مختلف دات نت تا بحال تهیه کرده و تقریبا هر موضوع جدیدی هم که اضافه می‌شود، بلافاصله یک سری جدید را تهیه می‌کنند. مدرسین انتخابی هم عموما افراد نامدار و باسوادی هستند.
پروژه‌ای رو در سایت کدپلکس شروع کردم جهت تهیه زیرنویس فارسی برای این ویدیوها:

http://dnps.codeplex.com/

این کار نسبت به کار تهیه زیرنویس‌های فارسی موجود برای فیلم‌های انگلیسی کار سخت‌تری است به چند دلیل:
- اسکریپت آماده‌ای وجود ندارد. کار شنیداری است.
- زمانبندی آماده‌ای وجود ندارد.
- مباحث تخصصی است.
- مدرس از ثانیه اول ویدیو تا ثانیه آخر آن حرف می‌زند!

برای مثال جهت تهیه زیرنویس‌های فارسی فیلم‌های انگلیسی عموما به سایت‌هایی مانند subscene.com مراجعه می‌شود. یک زیرنویس یا به قولی اسکریپت آماده یافت شده و شروع به ترجمه می‌شود. متن آماده است. زمانبندی آماده است و فقط کار ترجمه باقی می‌ماند.
اما در مورد ویدیوهای آموزشی انگلیسی خیر. به همین جهت در این زمینه کار آنچنانی تابحال صورت نگرفته.
کار زمانبری است. فعلا رکورد من برای هر سه دقیقه ویدیوی آموزشی که مدرس از ثانیه اول تا آخر آن حرف می‌زند، 40 دقیقه کار تهیه زیر نویس فارسی زمانبندی شده است.

سری جدیدی رو که شروع کردم تحت عنوان « Building Windows 8 Metro Apps in C# and XAML » در سایت pluralsight ارائه شده.
فعلا قسمت اول آن زیرنویس دار شده و از اینجا قابل دریافت است. برای مشاهده آن‌ها برنامه با کیفیت و رایگان KMPlayer توصیه می‌شود.

لیست ویدیوهای قسمت اول آن به شرح زیر است:

```
Building Windows 8 Metro Apps in C# and XAML
Overview  00:50:41
This modules provides an overview of how to develop Windows 8 Metro style applications in C# and XAML.

Introduction
XAML and Codebehind
Asynchronous APIs
Demo: Asynchronous APIs
Files and Networking
Demo: Requesting Capabilities
Integrating with Windows 8
WinRT and .NET
Summary
```

کلمه سی شارپ در این قسمت کمی غلط انداز است. بیشتر بحث و توضیح است تا کد نویسی. بنابراین برای عموم قابل استفاده است. خصوصا نگاهی دارد به تازه‌های ویندوز 8 از دیدگاه برنامه نویس‌ها مانند سطوح دسترسی برنامه‌های مترو، معرفی Charms ،

نحوه به اشتراک گذاری اطلاعات در بین برنامه‌های مترو نصب شده، برای مثال جایی که دیگر Clipboard سابق وجود ندارد و مواردی از این دست.

5 قسمت دیگر این مبحث باقیمانده که هر زمان تکمیل شد، خبرش رو خواهید شنید و تقریبا از این پس این سایت به همین ترتیب جلو خواهد رفت. فکر می‌کنم اینطوری مفیدتر باشد. هر از چندگاهی یک مبحث جدید زیرنویس دار شده را می‌توانید مشاهده کنید.

اگر علاقمند بودید می‌تونید در این پروژه شرکت کنید و یک موضوع جدید و مستقل را برای تهیه زیرنویس شروع کنید. یا حداقل اگر پس از مشاهده این سری آماده شده، اصلاحی را انجام دادید، می‌تونید اون رو برای اعمال در اینجا ارسال کنید.

نظرات خوانندگان

نویسنده: Sirwan Afifi
تاریخ: ۱۳۹۰/۱۰/۲۳ ۱۰:۳۴:۲۰

سلام استاد خیلی ممنون عالیه.
فقط من توی این سایت Building Windows 8 Metro Apps in C# and XAML لینک دانلودی نمی بینم.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۱۰/۲۳ ۱۰:۴۳:۳۹

من فقط لینک زیرنویس‌های تهیه شده یا به قولی کار خودم را اینجا ذکر کردم. یافتن اصل ویدیوها (که رایگان نیستند) در گوگل کار ساده‌ای است و این قسمت با خودتان است. علاقمند بودید آن‌ها را خریداری کنید. علاقمند بودید، دانلود کنید. برای مثال «win8-csxaml» را در گوگل جستجو کنید. در همان صفحه اول لینک‌های دانلود ظاهر می‌شوند.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۱۰/۲۴ ۰۰:۵۷:۳۷

جهت یادآوری:
من هر مطلبی که مرتبط به موضوع نباشد را حذف می‌کنم.
در مورد «زیرنویس»های تهیه شده مطلبی دارید؟
اگر خیر لطفا از سایت‌های دیگر استفاده کنید. با تشکر

نویسنده: Reza Bozorgi
تاریخ: ۱۳۹۰/۱۰/۲۵ ۰۱:۴۴:۳۳

- کار خیلی ارزشمندیه واقعا. و یه سبک نویی هست در تولید محتوا به زبان فارسی.
- توی support سایت گفتن که به زودی زیرنویس برا همه‌ی Course ها خواهیم گذاشت.
- اگر امکان براتون داشته باشه، یه پستی بزنین که نحوه‌ی تولید اسکریپت رو توش توضیح بدین. چون مشکل اصلی اینجاست.
- از اونجایی که فکر میکنم، این کار احتمالا دستی انجام میگیره(یعنی به راحتی نیست)
یه پیشنهاد دارم: به دلیل تایم‌های کوتاه هر section و بعد از اون هر part از section ها که خیلی کوتاهند. میشه تا اقدام رسمی خود پلورال‌سایت به تولید زیرنویس، فقط حرفهایی که در کلیپ توسط مدرس زده شده نوشته شود(بدون توجه به زمانبدی).
که این میتونه دوجور باشه:
- یکی از الف تا ی.
- یکی دیگه، صرفا اشاره و رساندن موضوع به صورت تیتر وار.
خب از مزیت‌های این کار:
- کاهش شدید زمانبری تولید محتواست.
- توان تولید محتوا(یا اصلاح محتوا) توسط افراد بیشتر.
- با توجه به ادای شفاف کلمات مدرسان، و با وجود همچین چیزی احتمالا اکثر مطالب قابل فهم‌تر خواهد بود.
- و شاید جالب‌تر از اینها، گزیده نوشته‌هایی خواهند شد، که برای مرور کردن دوره‌ها در آینده بسیار مفید و کارا خواهد بود.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۱۰/۲۵ ۰۸:۳۲:۰۰

- برای تولید زیرنویس از برنامه خودم استفاده می‌کنم: (^)
- کلا زیرنویس بدون زمانبندی ارزشی نداره و بیشتر شبیه به آش است تا کار. البته این نظر بنده است و بر همین اساس هم جلو خواهد رفت.

نویسنده: مجید رفیعی
تاریخ: ۱۳۹۰/۱۰/۲۵ ۰۹:۳۱:۲۸

جناب نصیری! دستتون درد نکنه ... روش بسیار جالبی در تولید محتوا به زبان فارسی است.

نویسنده: Reza Bozorgi
تاریخ: ۱۳۹۰/۱۰/۲۵ ۱۰:۰۷:۲۳

حرفتون درسته. زیرنویس، حتی اگه یک ثانیه هم عقب جلو بشه بی ارزشه. ولی پیشنهاد بنده چیز دیگری بود. که ظاهرا قصد اون‌کارو ندارید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۱۰/۲۵ ۱۰:۱۱:۴۶

عرض کردم. قصد ندارم آش درست کنم. قصد خاک بازی هم ندارم. هر کاری یک تعریفی داره.

نویسنده: PetekDincos
تاریخ: ۱۳۹۰/۱۰/۲۵ ۱۹:۲۹:۳۲

با سلام
جناب نصیری کارهای شما همیشه با ارزش بوده و این کارتون نیز خیلی با ارزشه ممنون و تشکر

نویسنده: امید آزادی
تاریخ: ۱۳۹۰/۱۰/۲۶ ۲۳:۱۹:۲۰

سلام.
خدا بهتون قوت بیشتر بده. فکرش رو که می کنم، ایده ی ناب و جالبی رو انتخاب کردید.

نویسنده: Mahdi Rad
تاریخ: ۱۳۹۰/۱۰/۲۸ ۱۳:۰۱:۰۹

با سلام و ووقت بخیر.
اول ممنون و بسیار ممنون از کار شما که بسیار ارزشمند است و این خود الگوییست برای اینچنین اشتراک گذاری اطلاعات که متاسفانه کمتر از هموطنان دیده می شود دارد.

اما اینکار هم وقت فراوانی می گیرد و هم خسته کننده است.
یک نظر. اگر توجه کرده باشید، اساتید در اکثریت این ویدیوهای آموزشی با لفظ بسیار ساده و بدون لهجه صحبت می کنند. نمیدونم تابحال از سرویس تشخیص گفتار و ترجمه یوتیوب استفاده کرده اید یا نه. خود کار فیلم رو برای شما زیرنویس و ترجمه می کند. این خیلی جالب است اگر بشود شروع کرد و چنین چیزی رو نوشت. البته بحث ترجمه به کنار. چون (فعلا) روبات های مترجم قدرت کافی در این زمینه را ندارند ولی همان بخش اول بسیار کار راه انداز است. فیلم را بگیرد. بر اساس زمان فیلم، جایی که گفتار شروع شد متن را بنویسد تا پایان آن مکالمه. و دوباره ... وقت برترین قسمت این نرمافزار همان تشخیص گفتار است که البته می توان از سرویس های ارائه شده مثل خود گوگل هم استفاده نمود.

عنوان: **زیر نویس فارسی ویدیوهای ساخت برنامه‌های مترو توسط سی شارپ و XAML - قسمت دوم**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۱۰/۲۵ ۱۰:۲۳:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: WinRT

زیرنویس‌های فارسی قسمت دوم « Building Windows 8 Metro Apps in C# and XAML » را از اینجا و یا اینجا می‌تونید دریافت کنید.

لیست سرفصل‌های قسمت دوم به شرح زیر است:

```
Layout 00:46:16
C# Metro applications have access to numerous XAML layout features.
This module describes those services, and shows how to use them to support Windows 8 features such as
display orientation, and snap.

Introduction
Layout System
Size Properties
Alignment
Margin
Demo: Margin and Alignment
Padding
Panels
Demo: Canvas
Demo: Grid and Snap
Data-Oriented Panels
ScrollViewer
Metro Layout Conventions
Layout Change Events
Summary
```

در کل این قسمت هم آنچنان کاری به برنامه نویسی ندارد و به بررسی و معرفی امکانات طرحبندی XAML می‌پردازد؛ به علاوه یک سری قراردادهای خاص مترو و همچنین نحوه‌ی کنار آمدن با حالت snapping ویژه ویندوز 8.

قسمت سوم مروری دارد بر کنترل‌های XAML که حدودا یک هفته دیگر زیرنویس‌های آن تمام خواهد شد.
لطفا اگر پس از مشاهده این سری آماده شده، اصلاحی را انجام دادید، اون رو برای اعمال در اینجا ارسال کنید. از این برنامه هم جهت ویرایش فایل‌ها می‌توان استفاده کرد.

نظرات خوانندگان

نویسنده: D_Ramezani
تاریخ: ۱۳۹۰/۱۰/۲۵ ۱۵:۲۲:۱۹

با سلام استاد در صورت امکان لینک خود فایل های تصویری هم قرار بدید . با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۱۰/۲۵ ۱۵:۵۸:۳۵

امکانش نیست؛ در مطلب قبلی توضیح دادم. این‌ها رایگان نیستند؛ باید خریداری شوند. یا هر روش دیگری که صلاح می‌دانید. من فقط کار خودم را به اشتراک گذاشتم.

نویسنده: Farhad Yazdan-Panah
تاریخ: ۱۳۹۰/۱۰/۲۵ ۲۳:۵۳:۰۷

با تشکر از کارهاتون و اشتراک گذاشتنشون!
(کاش لهجه ی Ian Griffiths رو هم اصلاح می کردید)

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۱۰/۲۶ ۰۹:۴۳:۵۱

در سری اولی که تهیه کردم یک واژه بود که به این صورت تلفظ می‌کرد: «لَت». در قسمت دوم متوجه شدم که منظورش همان «layout» است!
وبلاگش اینجا است: (^)

نویسنده: Mehdi Agazadeh
تاریخ: ۱۳۹۰/۱۰/۲۶ ۱۷:۰۹:۱۳

استاد همین کار رو ادامه بدید کارتون عالیه با تشکر

عنوان: **زیر نویس فارسی ویدیوهای ساخت برنامه‌های مترو توسط سی شارپ و XAML - قسمت سوم**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۱۰/۳۰ ۱۴:۰۸:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: WinRT

زیرنویس‌های فارسی قسمت سوم « Building Windows 8 Metro Apps in C# and XAML » را از اینجا و یا اینجا می‌تونید دریافت کنید.

لیست سرفصل‌های قسمت سوم به شرح زیر است:

```
Controls  00:48:08
WinRT provides numerous XAML controls, which act as the building blocks of your applications.
Some of these will be familiar, while some are new to Metro.
Introduction
Command Controls
Demo: Application Bar
Context Menus
Demo: Context Menus
Application Settings
CheckBox, RadioButton, and ToggleSwitch
Progress Controls
Demo: Progress Controls
Displaying Text
Demo: Displaying Text
Text Entry
Sliders
Styling
WebView
UserControl
Summary
```

مطالبی که بیشتر در این سری بر روی آن‌ها تمرکز شده، مباحث صفحات لمسی و تغییرات کنترل‌های ویندوز 8 جهت سازگاری با این مساله است. بنابراین در این سری بیشتر از هرچیزی tap ، finger و touch را خواهید شنید. جایی که چند جا را می‌شود همزمان لمس کرد و با چندین انگشت همزمان کار کرد که نکات قابل توجهی را نسبت به دنیای تک بعدی ماوس به همراه دارد.

نظرات خوانندگان

نویسنده: Farhad Yazdan-Panah
تاریخ: ۱۳۹۰/۱۰/۳۰ ۱۴:۳۵:۱۹

ممنون.

عنوان: **زیر نویس فارسی ویدیوهای ساخت برنامه‌های مترو توسط سی شارپ و XAML - قسمت چهارم**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۱۱/۰۶ ۰۰:۱۶:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: WinRT

زیرنویس‌های فارسی قسمت چهارم « Building Windows 8 Metro Apps in C# and XAML » را از اینجا و یا اینجا می‌تونید دریافت کنید.

لیست سرفصل‌های قسمت چهارم به شرح زیر است:

```
List Controls  00:31:14
This module shows the Metro controls available to XAML applications for working with collections of
items.
This includes the new GridView and ListView controls,
which are optimized for handling collections in a touch-based user interface.

Introduction
Items Controls
Demo: ListBox vs ListView
Demo: GridView
Demo: FlipView
Common ItemsControl
Semantic Zoom
Demo: JumpViewer
Summary
```

این قسمت به بررسی یک سری کنترل لیستی جدید ویندوز 8 اختصاص دارد شامل ListView بازنویسی شده و همچنین GridView به همراه دو کنترل FlipView و JumpViewer که تمام این‌ها جهت کار با صفحات لمسی بهینه سازی شده‌اند.

نظرات خوانندگان

نویسنده: Saba_2082
تاریخ: ۱۳۹۰/۱۱/۱۲ ۱۶:۳۷:۰۱

خود این ویدیو ها رو جه جوری داشته باشیم؟ همش پولیه

نویسنده: Naser Tahery
تاریخ: ۱۳۹۰/۱۱/۱۳ ۲۲:۴۴:۴۸

میتونید دانلود کنید. این عبارت رو جستجو کنید "win8-csxaml"
البته بهتره که بخرینش

عنوان: **زیر نویس فارسی ویدیوهای ساخت برنامه‌های مترو توسط سی شارپ و XAML - قسمت پنجم**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۱۱/۱۴ ۰۱:۰۴:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: WinRT

زیرنویس‌های فارسی قسمت پنجم « Building Windows 8 Metro Apps in C# and XAML » را از اینجا و یا اینجا می‌تونید دریافت کنید.

لیست سرفصل‌های قسمت پنجم به شرح زیر است:

```
Application Model  00:59:50
Metro and WinRT introduce some significant changes to the world in which applications execute.
This module describes the implications for developers.
Introduction
Application Lifecycle
Demo: Application Lifecycle
Managing State
Demo: Saving State
Splash Screens
Launching Applications
Application Manifest
Packaging
Summary
```

این قسمت به جزئیات نحوه اجرای برنامه‌های مترو می‌پردازد. اگر با IIS کار کرده باشید، سیکل اجرایی برنامه‌های مترو ویندوز 8، همانند سیکل اجرایی برنامه‌های ASP.NET شده است! ویندوز مختار است برنامه شما را پس از مدتی بیکاری (البته این مدت در اینجا فقط 5 ثانیه است!)، معلق کرده یا حتی خاتمه دهد و تمام این‌ها هم از دید کاربر نهایی مخفی است. مانند زمانیکه یک برنامه ASP.NET پس از مدتی بیکاری، توسط IIS خاتمه می‌یابد (از حافظه خارج می‌شود) و پس از مدتی با رسیدن یک درخواست جدید، یک پروسه جدید برای اجرای آن ایجاد شده و مجددا سایت شروع به کار خواهد کرد؛ اینجا هم در دنیای مترو تقریبا به همین نحو با یک برنامه رفتار می‌شود.
یک نکته جالب دیگر هم در برنامه‌های مترو وجود دارد: ترد اصلی برنامه از ترد رابط کاربری جدا شده است. برای مثال سازنده کلاس App برنامه در یک ترد و رابط کاربری برنامه در ترد مجزای دیگری اجرا می‌شوند.
به علاوه روش‌های متفاوتی هم برای اجرای برنامه‌های مترو درنظر گرفته شده. دیگر فقط حالت کلیک بر روی یک برنامه سبب اجرای آن نمی‌شود. می‌توان بر اساس اتصال یک سخت افزار خاص به سیستم یا حتی یک جستجو هم سبب اجرای برنامه‌ای شد. برای مثال می‌توانید برنامه خود را طوری طراحی کنید که نتیجه‌ی جستجویی را در سیستم نمایش دهد.
سیستم بسته بندی برنامه‌های مترو نیز بسیار شبیه به فایل‌های XAP برنامه‌های سیلورلایت است که همه چیز داخل یک فایل قرار داده می‌شود؛ از فایل‌های تنظیمات برنامه تا فایل‌های کامپایل شده و منابع مورد نیاز. البته در اینجا نامش به appx تغییر یافته است به علاوه یک فایل cer که حاوی مجوز دیجیتال اجباری توزیع برنامه‌های مترو در فروشگاه ویندوز است.

نظرات خوانندگان

نویسنده: Farhad Yazdan-Panah
تاریخ: ۰۱:۴۰:۱۲ ۱۳۹۰/۱۱/۱۴

ممنون.
البته فکر کنم که در ASP.NET4.0 میشه کاری کرد که هیچوقت برنامه (وبسایت) از حافظه خارج نشه.
در کل ایده مترو (و ایده های جدید مایکروسافت) نشون میده که یه خبرایی هست تو اون شرکت. ملت دارن یه کارایی می کنند.

نویسنده: وحید نصیری
تاریخ: ۰۸:۴۶:۰۷ ۱۳۹۰/۱۱/۱۴

حالت auto-start مربوط به ASP.NET 4 و البته IIS7 به بعد (^) ، مقدار همیشه در حال اجرا هم دارد؛ اما این مورد از تنظیمات Application pool تاثیر می‌پذیره و پس از recycle شدن برنامه مجددا برنامه رو اجرا می‌کنه به صورت خودکار. اما برنامه بر اساس تنظیمات Application pool، پس از یک مدت بیکاری حتما خاتمه خواهد یافت.
در مورد جزئیات کاری Application pool در IIS مطالبی در سایت هست : (^) و (^)

نویسنده: Farhad Yazdan-Panah
تاریخ: ۱۲:۱۰:۲۳ ۱۳۹۰/۱۱/۱۴

ممنون. عالی بود و جالب.

نویسنده: shahin kiassat
تاریخ: ۱۴:۳۰:۳۵ ۱۳۹۰/۱۱/۱۴

سلام.
آقای نصیری می خواستم یک تشکر ویژه و یک خسته نباشید خدمتتون عرض کنم بابت زحماتی که می کشید.
جدا از زیرنویس ها که اخیرا شروع به تولید کردید و ایده ی بسیار جالبی هست هر زمان یکی از مطالب وبلاگتون رو مطالعه کردم خیلی نکته ها یاد گرفتم و یک میان بر فوق العاده بوده.
ارادت مند.

عنوان: **زیر نویس فارسی ویدیوهای ساخت برنامه‌های مترو توسط سی شارپ و XAML - قسمت آخر**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۱۱/۲۵ ۰۰:۰۸:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: WinRT

زیرنویس‌های فارسی قسمت آخر « Building Windows 8 Metro Apps in C# and XAML » را از اینجا می‌تونید دریافت کنید.

این قسمت برای کسانی که می‌خواهند مروری بر مفاهیم Binding موجود در WPF و سیلورلایت داشته باشند و همچنین تفاوت‌های آن‌را با نمونه موجود در WinRT بررسی کنند، بسیار مفید است. در کل سیستم Binding موجود در WinRT یک نمونه ساده شده آنچیزی است که در سیلورلایت موجود است (و البته سیلورلایت هم یک نمونه ساده شده WPF است!).
برای مثال خاصیت UpdateSourceTrigger موجود در عبارات Binding مرتبط با WPF و سیلورلایت فعلا در WinRT وجود ندارد.
تفاوت دیگر این است که هرچند اینترفیس INotifyPropertyChanged در WinRT هم وجود دارد اما نمونه مهیای در فضای نام جدیدی به نام windows.ui.xaml.data توسط WinRT شناسایی می‌شود و اگر کدهای سایر فناوری‌های مشابه را به سیستم مترو تبدیل کنید، کار نخواهند کرد! چون این اینترفیس پیشتر در فضای نام System.ComponentModel تعریف شده بود و هنوز هم حضور دارد (فقط در حد یک تغییر سطر تعاریف فضاهای نام کفایت می‌کند).
یک تله دیگر هم در WinRT وجود دارد. در اینجا هم کلاسی به نام DependencyObject وجود دارد که ... معادل نمونه مشابه WPF و Silverlight نیست و XAML امکان تشخیص خودکار تغییرات خواص آن‌را ندارد.
همچنین اینترفیس INotifyCollectionChanged مرتبط با ObservableCollection موجود در WPF و Silverlight در WinRT فعلا وجود خارجی ندارند (هرچند هنوز در خود دات نت فریم ورک وجود دارند، اما کار نمی‌کنند). به نظر قرار است تا قبل از ارائه نهایی ویندوز 8 در این مورد تصمیم گیری شود. اما در اینجا باید از IObservableVector استفاده کرد! (کلا این کلمه Vector ابراز وجود زاید طراحان سی++ مترو است! یعنی ما هنوز هم زنده‌ایم و سی++ هنوز هم مهم است!)
نحوه گروه بندی اطلاعات نیز تغییر کرده است و باید منبع داده‌ای، اینترفیس جدید IGroupInfo را پیاده سازی کند. به علاوه CollectionViewSource آن نیز فعلا قابلیت‌های جستجو و مرتب سازی موجود در WPF و سیلورلایت را ارائه نمی‌دهد.

نظرات خوانندگان

نویسنده: A. Karimi
تاریخ: ۱۳۹۰/۱۱/۲۵ ۲۳:۴۵:۵۷

امیدوارم WinRT همینطوری نمونه

عنوان: **معرفی Lex.Db**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۸/۲۹ ۱:۰
آدرس: www.dotnettips.info
گروه‌ها: Silverlight, WinRT, Lex.Db, Embedded Database

Lex.Db یک بانک اطلاعاتی درون پروسه‌ای (مدفون شده یا embedded) بسیار سریع نوشته شده با سی‌شارپ است. این بانک اطلاعاتی کم حجم، سورس باز بوده و مجوز استفاده از آن LGPL است. به این معنا که استفاده از اسمبلی‌های آن در هر نوع پروژه‌ای آزاد است.
نکته مهم آن سازگاری با برنامه‌های دات نت 4 به بعد، همچنین برنامه‌های ویندوز 8، سیلورلایت 5، ویندوز فون 8 و همچنین اندروید (از طریق Mono) است. به علاوه چون با دات نت تهیه شده است، دیگر نیازی نیست دو نگارش 32 بیتی و 64 بیتی آن توزیع شوند و به این ترتیب مشکلات توزیع بانک‌های اطلاعاتی native مانند SQLite را ندارد (و مطابق ادعای نویسنده آلمانی آن، از SQLite سریعتر است).
API این بانک اطلاعاتی، هر دو نوع متدهای synchronous و asynchronous را شامل می‌شود؛ به همین جهت با برنامه‌های ویندوز 8 و سیلورلایت نیز سازگاری دارد.
Lex.Db از برنامه‌های چندریسمانی و همچنین استفاده از یک بانک اطلاعاتی آن توسط چندین پروسه همزمان نیز پشتیبانی می‌کند.
در ادامه مروری خواهیم داشت بر نحوه استفاده از آن در حالت طراحی رابطه‌ای؛ از این جهت که فعلا به ظاهر این بانک اطلاعاتی روابط را پشتیبانی نمی‌کند، اما در عمل پیاده سازی آن مشکل نیست.

دریافت Lex.Db

برای دریافت Lex.Db، دستور ذیل را در خط فرمان پاورشل نیوگت وارد نمائید:

```
PM> Install-Package Lex.Db
```

بسته به نوع پروژه شما (دات نت یا WinRT یا ...)، اسمبلی متناسبی به پروژه اضافه خواهد شد.

مدل‌های برنامه

```
public class Product
    {
        public int Id { get; set; }
        public string Name { get; set; }
    }

    public class Customer
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public string City { get; set; }
    }

    public class Order
    {
        public int Id { get; set; }
        public int? CustomerFK { get; set; }
        public int[] ProductsFK { get; set; }
    }
```

مدل‌های برنامه آزمایشی مطلب جاری را در اینجا ملاحظه می‌کنید. برای طراحی روابط یک به صفر یا یک و همچنین یک به چند، تنها کافی است کلیدهای اصلی یا آرایه‌ای از کلیدهای اصلی مرتبط را در اینجا ذخیره کنیم، که نمونه‌ای از آن‌را در کلاس Order ملاحظه می‌کنید.

آغاز بانک اطلاعاتی

```
public static class Database
    {
        public static DbInstance Instance { get; private set; }

        public static DbTable<Product> Products { get; private set; }
        public static DbTable<Order> Orders { get; private set; }
        public static DbTable<Customer> Customers { get; private set; }

        /// <summary>
        /// سازنده استاتیکی که در طول عمر برنامه فقط یکبار اجرا می‌شود
        /// </summary>
        static Database()
        {
            createDb();
            getTables();
        }

        private static void getTables()
        {
            Products = Instance.Table<Product>();
            Customers = Instance.Table<Customer>();
            Orders = Instance.Table<Order>();
        }

        private static void createDb()
        {
            Instance = new DbInstance(Path.Combine(Environment.CurrentDirectory, "LexDbTests"));

            Instance.Map<Product>()
                    .WithIndex("NameIdx", x => x.Name)
                    .Automap(i => i.Id, true);

            Instance.Map<Order>()
                    .Automap(i => i.Id, true);

            Instance.Map<Customer>()
                    .WithIndex("NameIdx", x => x.Name)
                    .WithIndex("CityIdx", x => x.City)
                    .Automap(i => i.Id, true);

            Instance.Initialize();
        }
    }
```

کلاس دیتابیس و سازنده آن، استاتیک تعریف شده‌اند؛ تا در طول عمر برنامه تنها یکبار وهله سازی شوند. new DbInstance یک وهله جدید از بانک اطلاعاتی را آغاز می‌کند. سازنده آن، مسیر پوشه‌ای که فایل‌های این بانک اطلاعاتی در آن ذخیره خواهند شد را دریافت می‌کند. Lex.Db به ازای هر کلاس مدلی که به آن معرفی شود، دو فایل data و index را ایجاد می‌کند.

سپس توسط وهله‌ای از بانک اطلاعاتی که ایجاد کردیم، کار معرفی خواص مدل‌های برنامه توسط متد Map و Automap انجام می‌شود. متد Automap خاصیت primary key کلاس را دریافت کرده و همچنین پارامتر دوم آن مشخص می‌کند که آیا این کلید اصلی به صورت خودکار ایجاد شود یا خیر. به علاوه در همینجا می‌توان روی فیلدهای مختلف، ایندکس نیز ایجاد کرد. متد WithIndex یک نام دلخواه را دریافت کرده و سپس خاصیتی را که باید بر روی آن ایندکس ایجاد شود، دریافت می‌کند.

در نهایت متد Initialize باید فراخوانی گردد. البته اگر برنامه شما WinRT است، این متد *Async* Initialize خواهد بود.

جداول نیز بر اساس مدل‌های برنامه از طریق متد Instance.Table در دسترس قرار گرفته‌اند.

افزودن اطلاعات به بانک اطلاعاتی

```
private static void addData()
        {
            var customer1 = new Customer { Name = "customer1", City = "City1" };
            var customer2 = new Customer { Name = "customer2", City = "City2" };
            Database.Instance.Save(customer1, customer2); // automatic Id assignment after Save

            var product1 = new Product { Name = "product1" };
            var product2 = new Product { Name = "product2" };
            Database.Instance.Save(product1, product2); // automatic Id assignment after Save

            var order1 = new Order { CustomerFK = customer1.Id, ProductsFK = new[] { product1.Id } };
            var order2 = new Order { CustomerFK = customer2.Id, ProductsFK = new[] { product1.Id,
product2.Id } };
            Database.Instance.Save(order1, order2); // automatic Id assignment after Save
```

```
        }
```

اکنون که کار آغاز بانک اطلاعاتی صورت گرفت، برای افزودن اطلاعات از متد Database.Instance.Save می‌توان استفاده کرد (در برنامه‌های WinRT از متد *Async* Save استفاده کنید).
در اینجا نیازی به ذکر Id نمونه‌های ساخته شده نیست؛ از این جهت که در حین عملیات Save، به صورت خودکار انتساب خواهند یافت.
همچنین نحوه مقدار دهی کلیدهای خارجی نیز با استفاده از همین کلیدهای اصلی آماده شده است.

واکشی تمام اطلاعات

```
private static void loadAll()
        {
            var orders = Database.Orders.LoadAll();
            foreach (var order in orders)
            {
                // نحوه دریافت اطلاعات مشتری بر اساس کلید خارجی ثبت شده
                var orderCustomer = Database.Customers.LoadByKey(order.CustomerFK.Value);
                Console.WriteLine("Order Id: {0}, Customer: {1} ({2}) {3}", order.Id,
orderCustomer.Name, orderCustomer.Id, orderCustomer.City);

                // نحوه بازیابی لیستی از اشیاء مرتبط از طریق آرایه‌ای از کلیدهای خارجی ثبت شده
                var orderProducts = Database.Products.LoadByKeys(order.ProductsFK);
                foreach (var product in orderProducts)
                {
                    Console.WriteLine("  Product Id: {0}, Name: {1}", product.Id, product.Name);
                }
            }
        }
```

بانک اطلاعاتی آغاز شد؛ تعدادی رکورد نیز در آن ثبت گردید. اکنون برای بازیابی اطلاعات می‌توان از متدهای در دسترس جداول کلاس Database استفاده کرد. برای مثال متد LoadAll تمام رکوردهای یک جدول را واکشی می‌کند (در برنامه‌های WinRT این متد LoadAll *Async* خواهد بود).
سپس با استفاده از متدهای LoadByKey و LoadByKeys، به سادگی می‌توان اشیاء مرتبط با هر سفارش را نیز واکشی کرد.

استفاده از ایندکس‌ها برای کوئری گرفتن

```
private static void queryingByAnIndex()
        {
            var name = "customer1";
            var customersList = Database.Customers
                                        .IndexQueryByKey("NameIdx", name)
                                        .ToList();
            foreach (var person in customersList)
            {
                Console.WriteLine(person.Name);
            }
        }
```

در ابتدای بحث، توسط متد WithIndex، تعدادی ایندکس را نیز تعریف کردیم. اکنون توسط این ایندکس‌ها و متد IndexQueryByKey، می‌توان کوئری‌هایی بسیار سریع را تهیه کرد.

```
// Using Take and Skip
            var list1 = Database.Orders.Query<int>() // primary idx
                                       .Take(1).Skip(2).ToList();

            // Querying Between Ranges
            var list2 = Database.Customers
                                .IndexQuery<string>("NameIdx")
                                .GreaterThan("a", orEqual: true).LessThan("d").ToList();
```

همچنین در اینجا متدهایی مانند Take و Skip و یا جستجو در یک بازه توسط متدهای GreaterThan و LessThan نیز پشتیبانی می‌شوند.

حذف رکوردها

```
private static void deletingRecords()
        {
            Database.Customers.DeleteByKey(key: 1);

            var customers = Database.Customers.LoadByKeys(new[] { 1, 2 });
            Database.Customers.Delete(customers);
        }
```

برای حذف رکوردها از متدهای DeleteByKey و یا Delete می‌توان استفاده کرد. متد Delete می‌تواند آرایه‌ای از اشیاء را نیز قبول کند.
و اگر خواستید کل بانک اطلاعاتی را خالی کنید، متد Database.Instance.Purge اینکار را انجام خواهد داد.

کدهای کامل این مثال را از اینجا نیز می‌توانید دریافت کنید:
Program-LexDb.cs

نظرات خوانندگان

نویسنده: ناصر طاهری
تاریخ: ۱۳۹۲/۰۸/۲۹ ۱۳:۲

ممنون از آموزشتون. آیا متدی مثل include هم داره ؟ یا باید مثل قسمت واکشی اطلاعات در همین آموزش عمل کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۸/۲۹ ۱۳:۱۵

lazy loading و eager loading ندارد. مثل همین مثال باید عمل کرد. هرجایی که نیاز است، خودتان باید اطلاعات را واکشی کنید. cascade delete هم ندارد. اگر لازم هست، خودتان دستی لیست Idها را بدهید تا حذف کند.

نویسنده: علیرضا
تاریخ: ۱۳۹۲/۰۸/۲۹ ۲۰:۵۹

و غیر از سرعت و مسئله 64/32 بیتی چه مزیتی نسبت به SQLite داره؟

نویسنده: خوزستان
تاریخ: ۱۳۹۲/۰۸/۲۹ ۲۲:۵۷

این بانک اطلاعاتی چون بصورت embeded هست سرعت لود اطلاعاتش نسبت با بانکهای دیگر بالاست ؟

نویسنده: سوین
تاریخ: ۱۳۹۲/۰۹/۰۲ ۰:۴۴

سلام
آیا از TransactionScope به صورت کامل پشتیبانی می‌کنه ؟ SqlLite به طور کامل پشتیبانی نمی‌کرد .

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۰۲ ۱:۶

- Lex.Db در حقیقت یک بانک اطلاعاتی NoSQL است. مثال رابطه‌ای رو که من در اینجا نوشتم، فقط یک شبیه سازی روابط است.
- به صورت توکار با استفاده از قفل گذاری توسط کلاس ReaderWriterLockSlim آن، خواندن‌ها و نوشتن‌های همزمان توسط چندین ترد را مدیریت می‌کند. یعنی نیازی نیست کار اضافه‌تری از این لحاظ توسط استفاده کننده انجام شود. (SQLite برای این مساله نیاز به پیاده سازی اضافی دارد و نمی‌شود با آن در حالت معمول از طریق چندین ترد همزمان کار کرد)
- از الگوریتم RedBlackTree برای ایندکس گذاری و جستجو استفاده می‌کند.

نویسنده: مهدی
تاریخ: ۱۳۹۲/۰۹/۲۶ ۲۳:۲۶

با سلام؛ آیا از lex.db در حالت چند کاربره می‌توان استفاده کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۹/۲۶ ۲۳:۵۷

Lex.DB supports concurrent database access

عنوان: **تبدیل PDF به تصویر با استفاده از API توکار Window 8.1 در برنامه‌های غیر مترو دات نت**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۰۱ ۱۴:۵۵
آدرس: www.dotnettips.info
گروه‌ها: WinRT, PdfReport, PDF

ویندوز 8.1 دارای امکانات و API توکاری جهت نمایش و خواندن فایل‌های PDF در برنامه‌های مترو است. در ادامه قصد داریم از این امکانات در یک برنامه‌ی متداول دات نت، برای مثال یک برنامه‌ی کنسول غیر مترو استفاده کنیم.

آماده سازی برنامه‌های دات نت برای دسترسی به API مترو ویندوز 8.1

ابتدا یک برنامه‌ی کنسول دات نت 4.5.1 را آغاز کنید. برای دسترسی به API ویندوز 8.1 حتما نیاز است که حداقل از دات نت 4.5.1 شروع کرد. سپس برنامه را در VS.NET بسته و فایل پروژه آن‌را در یک ادیتور متنی باز کنید.
در ابتدای فایل csproj، نیاز است سطر TargetPlatformVersion ذیل اضافه شود.

```
<PropertyGroup>
    <TargetFrameworkVersion>v4.5.1</TargetFrameworkVersion>
    <TargetPlatformVersion>8.1</TargetPlatformVersion>
  </PropertyGroup>
```

سپس در همین فایل، ارجاعات زیر را نیز اضافه نمائید:

```
<ItemGroup>
    <Reference Include="System" />
    <Reference Include="System.ComponentModel.DataAnnotations" />
    <Reference Include="System.Core" />
    <Reference Include="System.ObjectModel" />
    <Reference Include="System.Xml.Linq" />
    <Reference Include="System.Data.DataSetExtensions" />
    <Reference Include="Microsoft.CSharp" />
    <Reference Include="System.Data" />
    <Reference Include="System.Xml" />
    <Reference Include="System.Threading" />
    <Reference Include="System.Threading.Tasks" />
  </ItemGroup>
  <ItemGroup>
    <Reference Include="Windows" />
    <Reference Include="System.Runtime" />
    <Reference Include="System.Runtime.WindowsRuntime" />
  </ItemGroup>
```

مواردی مانند System.Runtime، System.Runtime.WindowsRuntime امکان دسترسی به API ویندوز 8 را در برنامه‌های دات نت میسر می‌کنند.

یک نکته

اگر می‌خواهید این فرآیند را ساده و خودکار کنید، از قالب‌های پروژه‌ی مخصوص DesktopWinRT.Templates.vsix استفاده نمائید.
DesktopWinRT.Templates.vsix

افزودن ارجاعی به Nito.AsyncEx

چون برنامه‌ی مورد استفاده کنسول است و API ویندوز 8 کاملا async طراحی شده‌است، نیاز است با کمک AsyncContext موجود در کتابخانه‌ی Nito.AsyncEx بتوان از امکانات async و await در متد Main برنامه استفاده کرد. البته اگر از سایر برنامه‌های دسکتاپ استفاده می‌کنید، فقط کافی است امضای متد رخدادن گردان را به async تغییر دهید.

```
install-package Nito.AsyncEx
```

تبدیل استریم‌های دات نت به استریم‌های WinRT

اکثر متدهای WinRT با استریم‌هایی از نوع IRandomAccessStream کار می‌کنند. برای اینکه بتوان استریم استاندارد دات نت را به این نوع تبدیل کرد، می‌توان از کلاس‌های ذیل کمک گرفت:

```
using System;
using System.IO;
using Windows.Storage.Streams;

namespace ConsoleWin81PdfApiTest
{
    public static class MicrosoftStreamExtensions
    {
        public static IRandomAccessStream AsRandomAccessStream(this Stream stream)
        {
            return new RandomStream(stream);
        }

    }

    class RandomStream : IRandomAccessStream
    {
        readonly Stream _internstream;

        public RandomStream(Stream underlyingstream)
        {
            _internstream = underlyingstream;
        }

        public IInputStream GetInputStreamAt(ulong position)
        {
            _internstream.Position = (long)position;
            return _internstream.AsInputStream();
        }

        public IOutputStream GetOutputStreamAt(ulong position)
        {
            _internstream.Position = (long)position;
            return _internstream.AsOutputStream();
        }

        public ulong Size
        {
            get
            {
                return (ulong)_internstream.Length;
            }
            set
            {
                _internstream.SetLength((long)value);
            }
        }

        public bool CanRead
        {
            get { return _internstream.CanRead; }
        }

        public bool CanWrite
        {
            get { return _internstream.CanWrite; }
        }

        public IRandomAccessStream CloneStream()
        {
            throw new NotSupportedException();
        }

        public ulong Position
        {
            get { return (ulong)_internstream.Position; }
        }

        public void Seek(ulong position)
        {
            _internstream.Seek((long)position, SeekOrigin.Begin);
        }
```

```
        public void Dispose()
        {
            _internstream.Dispose();
        }

        public Windows.Foundation.IAsyncOperationWithProgress<IBuffer, uint> ReadAsync(IBuffer buffer,
uint count, InputStreamOptions options)
        {
            return GetInputStreamAt(Position).ReadAsync(buffer, count, options);
        }

        public Windows.Foundation.IAsyncOperation<bool> FlushAsync()
        {
            return GetOutputStreamAt(Position).FlushAsync();
        }

        public Windows.Foundation.IAsyncOperationWithProgress<uint, uint> WriteAsync(IBuffer buffer)
        {
            return GetOutputStreamAt(Position).WriteAsync(buffer);
        }
    }
}
```

تا اینجا به یک متد الحاقی جدیدی به نام AsRandomAccessStream می‌رسیم که امکان تبدیل استریم استاندارد دات نت را به IRandomAccessStream مخصوص WinRT دارد. از آن می‌توان برای باز کردن یک فایل و ارسال استریم آن به توابع WinRT و یا ثبت استریم WinRT در یک فایل استفاده کرد.

خواندن فایل‌های PDF و تبدیل صفحات آن‌ها به تصویر

در ادامه کد کامل استفاده از API جدید ویندوز 8.1 را جهت خواندن فایل‌های PDF ملاحظه می‌کنید. این امکانات جدید در فضای نام Windows.Data.Pdf قرار دارند و صرفا امکان خواندن فایل‌های PDF را تدارک دیده‌اند.

```
using System;
using System.IO;
using System.Threading.Tasks;
using Windows.Data.Pdf;
using Nito.AsyncEx;

namespace ConsoleWin81PdfApiTest
{
    class Program
    {
        static void Main(string[] args)
        {
            AsyncContext.Run(async () =>
            {
                await test();
            });
        }

        private static async Task test()
        {
            using (var randomAccessStream = File.Open("PieChartPdfReport.pdf",
FileMode.Open).AsRandomAccessStream())
            {
                var pdfDocument = await PdfDocument.LoadFromStreamAsync(randomAccessStream);
                for (uint i = 0; i < pdfDocument.PageCount; i++)
                {
                    using (var page = pdfDocument.GetPage(i))
                    {
                        /*var renderOptions = new PdfPageRenderOptions
                        {
                            BackgroundColor = Colors.LightGray,
                            DestinationHeight = (uint) (page.Size.Height*10)
                        };*/

                        using (var stream = File.Open(string.Format("page-{0}.png", i + 1),
FileMode.OpenOrCreate).AsRandomAccessStream())
                        {
                            await page.RenderToStreamAsync(stream/*, renderOptions*/);
                            await stream.FlushAsync();
                        }
                    }
```

```
                }
            }
        }
    }
}
```

توضیحات:

- متد AsyncContext.Run جزو امکانات Nito.AsyncEx است و امکان نوشتن کدهای await دار را در متد Main یک برنامه‌ی کنسول فراهم می‌کند.

- متد File.Openدات نت، خروجی از نوع استریم دارد. برای تبدیل آن به نوع IRandomAccessStream، از متد الحاقی AsRandomAccessStream که پیشتر تهیه کردیم، می‌توان استفاده کرد.

- در ادامه متد PdfDocument.LoadFromStreamAsync این استریم خاص را دریافت کرده و امکان دسترسی به API ویندوز 8.1 را میسر می‌کند.

- توسط متد pdfDocument.GetPage می‌توان به صفحات مختلف فایل PDF باز شده دسترسی یافت. در اینجا متد page.RenderToStreamAsync، سبب رندر شدن صفحه با فرمت PNG می‌شود. این خروجی نهایتا باید در یک استریم از نوع IRandomAccessStream ثبت شود. در اینجا نیز می‌توان از متد File.Open در حالت FileMode.OpenOrCreate استفاده کرد.

- اگر می‌خواهید ابعاد تصویر نهایی و ویژگی‌های آن‌را تغییر دهید، می‌توان از پارامتر دوم متد page.RenderToStreamAsync استفاده کرد که شیءایی از نوع PdfPageRenderOptions را می‌پذیرد.

کدهای کامل این پروژه را از اینجا می‌توانید دریافت کنید

MicrosoftStreamExtensions.zip

برای مطالعه بیشتر

How to use specific WinRT API from Desktop apps

How to call WinRT APIs from .NET desktop apps

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۰۳ ۱۰:۴۹

یک نکته‌ی تکمیلی

اگر با استفاده از مطالب فوق قصد داشته باشید در WPF یک PDF Viewer درست کنید، می‌توان از متد ذیل استفاده کرد:

```
private async Task<List<System.Windows.Media.Imaging.BitmapImage>> getPdfPageImages()
        {
            var results = new List<System.Windows.Media.Imaging.BitmapImage>();
            using (var randomAccessStream = File.Open("PieChartPdfReport.pdf",
FileMode.Open).AsRandomAccessStream())
            {
                var pdfDocument = await PdfDocument.LoadFromStreamAsync(randomAccessStream);
                for (uint i = 0; i < pdfDocument.PageCount; i++)
                {
                    using (var memoryStream = new MemoryStream())
                    {
                        using (var stream = memoryStream.AsRandomAccessStream())
                        {
                            using (var page = pdfDocument.GetPage(i))
                            {
                                // Set render options
                                var renderOptions = new PdfPageRenderOptions
                                {
                                    BackgroundColor = Colors.LightGray,
                                    DestinationHeight = (uint)(page.Size.Height * 10)
                                };

                                await page.RenderToStreamAsync(stream); //, renderOptions);
                                await stream.FlushAsync();

                                var bitmapImage = new System.Windows.Media.Imaging.BitmapImage();
                                bitmapImage.BeginInit();
                                //Without this, BitmapImage uses lazy initialization by default and the
stream will be closed by then.
                                bitmapImage.CacheOption =
System.Windows.Media.Imaging.BitmapCacheOption.OnLoad;
                                bitmapImage.StreamSource = memoryStream;
                                bitmapImage.EndInit();

                                results.Add(bitmapImage);
                            }
                        }
                    }
                }
            }
            return results;
        }
```

بعد برای استفاده از BitmapImage های حاصل از آن، برای مثال نمایش اولین صفحه در یک کنترل Image استاندارد، می‌توان نوشت:

```
private async void Button_Click(object sender, RoutedEventArgs e)
        {
            var images = await this.getPdfPageImages();
            ImagePdf.Source = images.First();
        }
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۰۹ ۱۴:۵۱

استفاده از این نکته برای ساخت یک PDF Viewer ساده در WPF.