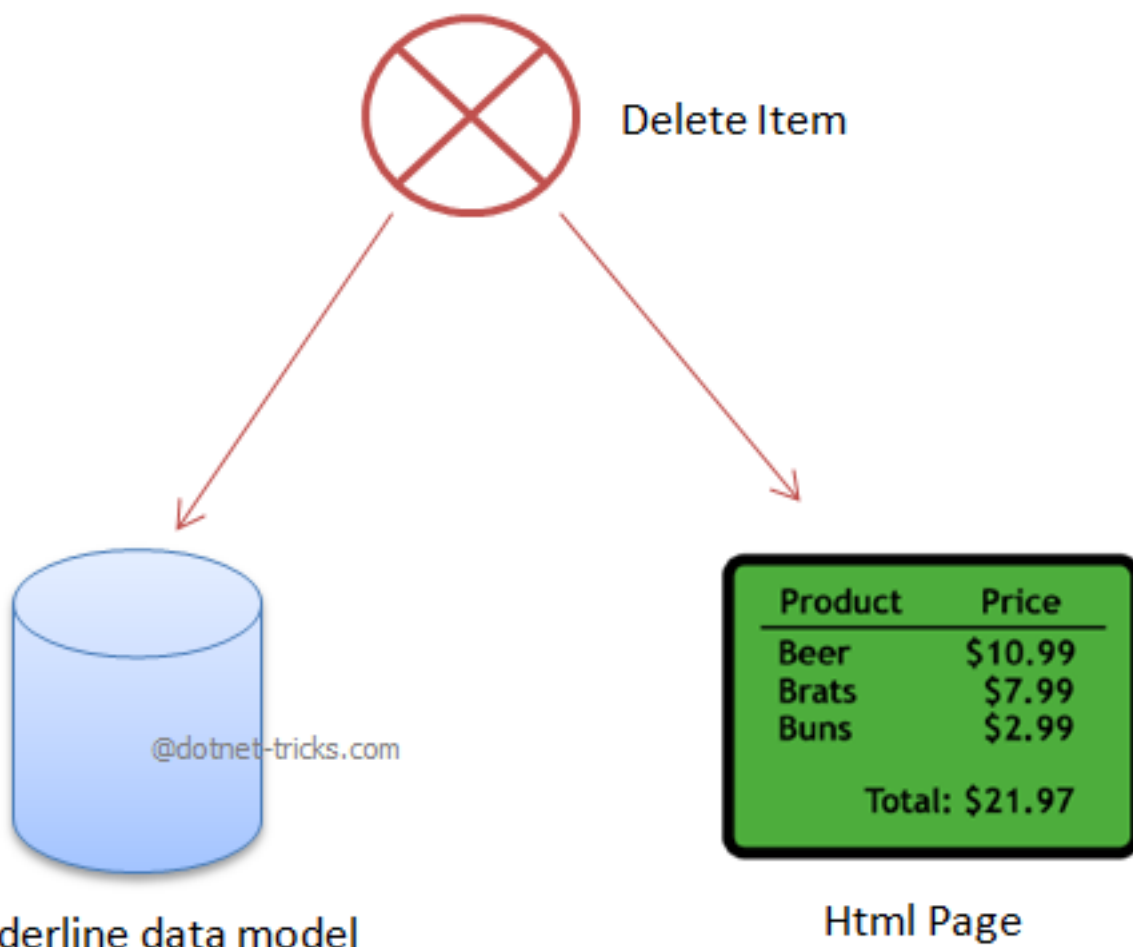


اگر از برنامه نویسی‌های پروژه‌های WPF درباره ویژگی‌های مهم الگوی MVVM بپرسید به احتمال زیاد اولین مطلبی که عنوان می‌شود این است که هنگام کار با الگوی MVVM در WPF باید از مباحث [data-binding](#) استفاده شود. به صورت خلاصه، [data-binding](#) مکانیزمی است که عناصر موجود در Xaml را به آبجکت‌های موجود در ViewModel یا سایر عناصر Xaml مقید می‌کند به طوری که با تغییر مقدار در آبجکت‌های ViewModel، عناصر View نیز خود را به روز می‌کنند یا با تغییر در مقادیر عناصر Xaml، آبجکت‌های متناظر در ViewModel نیز تغییر خواهند کرد (در صورت تنظیم Mode = TwoWay).

Knockout.js چیست؟

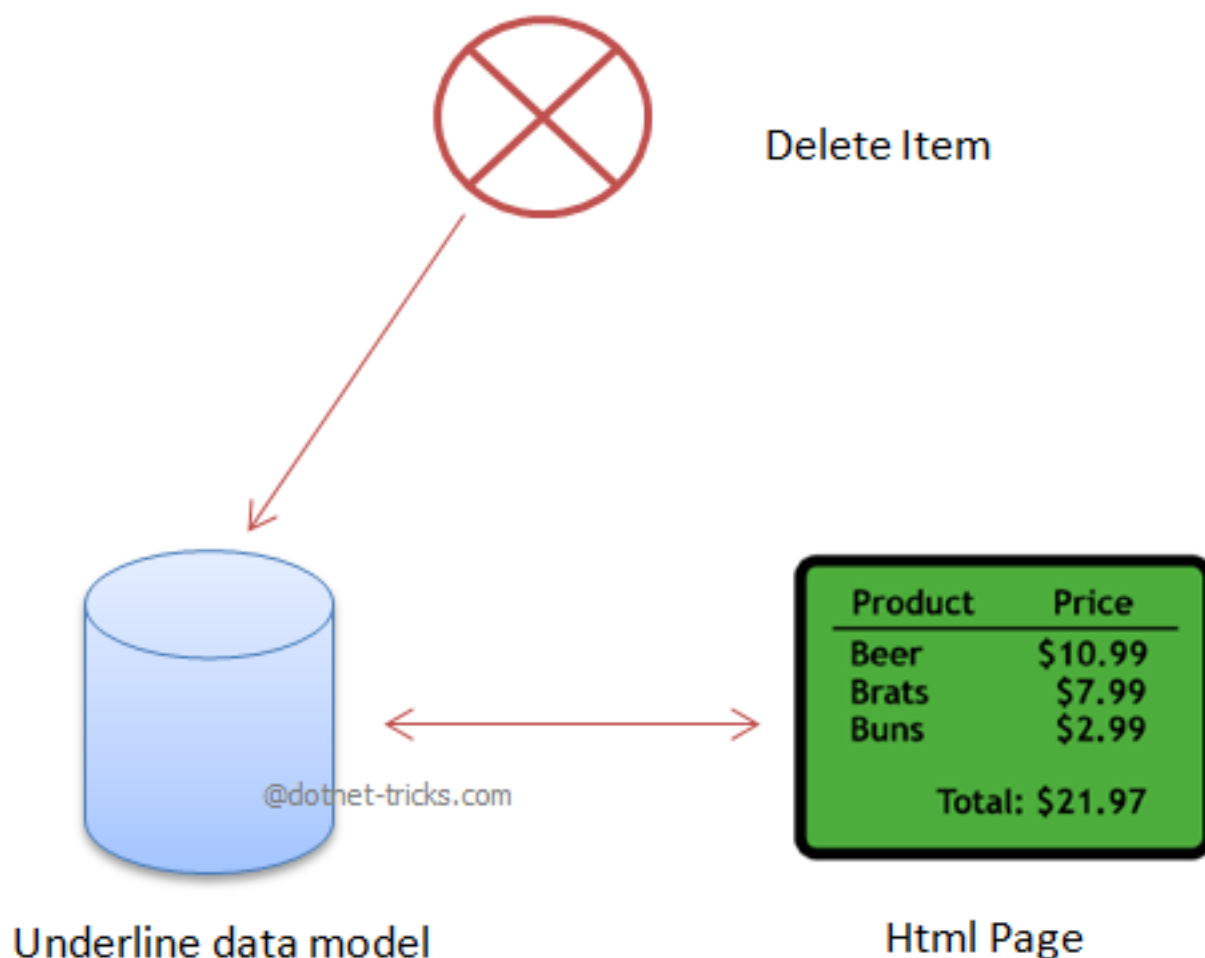
در یک جمله Knockout.js یک فریم ورک جاوا اسکریپت است که امکان پیاده سازی الگوی MVVM و مکانیزم [data-binding](#) را در پروژه‌های تحت وب به راحتی میسر می‌کند. به عبارت دیگر عناصر DOM را به [data-model](#) و آبجکت‌های [data-model](#) را به عناصر DOM مقید می‌کند، به طوری که با هر تغییر در مقدار یا وضعیت این عناصر یا آبجکت‌ها، تغییرات به موارد مقید شده نیز اعمال می‌گردد. به تصاویر زیر دقت کنید!

به روز رسانی [data-model](#) بدون استفاده از KO



Before Knockout: Manually tracking dependencies between HTML elements and their underlying data

به روز رسانی data-model با استفاده از KO



After Knockout: Automatically tracking dependencies between HTML elements and their underlying data

ویژگی‌های مهم KO

«ارائه یک راه حل بسیار ساده و واضح برای اتصال بخش‌های مختلف UI به data-model به روز رسانی خودکار عناصر و بخش‌های مختلف UI بر اساس تغییرات صورت گرفته در data-model به صورت کامل با کتابخانه و توابع javascript پیاده سازی شده است. حجم بسیار کم (سیزده کیلو بایت) بعد از فشرده سازی سازگار با تمام مرورگرهای جدید (IE 6+, Firefox 2+, Chrome, Safari, ...) امکان استفاده راحت بدون اعمال تغییرات اساسی در معماری پروژه هایی که در فاز توسعه هستند و بخشی از مسیر توسعه را طی کرده اند و...»

آیا KO برای تکمیل JQuery در نظر گرفته شده است یا جایگزین؟

در اینکه JQuery بسیار محبوب است و در اکثر پروژه‌های تحت وب مورد استفاده است شکی وجود ندارد ولی این بدان معنی

نیست که با توجه به وجود JQuery و محبوبیت آن دیگر نیازی به KO احساس نمی‌شود. به عنوان یک مثال ساده : فرض کنید در یک قسمت از پروژه قصد داریم یک لیست از داده‌ها را نمایش دهیم. در پایین لیست تعداد آیتم‌های موجود در لیست مورد نظر نمایش داده می‌شود. یک دکمه Add داریم که امکان اضافه شدن آیتم جدید را در اختیار ما قرار می‌دهد. بعد از اضافه شدن یک مقدار، باید عددی که تعداد آیتم‌های لیست را نمایش می‌دهد به روز کنیم. خب اگر قصد داشته باشیم این کار را با JQuery انجام دهیم راه حل‌های زیر پیش رو است :

« به دست آوردن تعداد trهای جدول موجود؛

« به دست آوردن تعداد divهای موجود با استفاده از یک کلاس مشخص css؛

« یا حتی به دست آوردن تعداد آیتم‌های نمایشی در span هایی مشخص.

و البته سایر راه حل‌ها...

حال فرض کنید دکمه‌های دیگر نظیر Delete نیز مد نظر باشد که مراحل بالا تکرار خواهند شد. اما با استفاده از KO به راحتی می‌توانیم تعداد آیتم‌های موجود در یک آرایه را به یک عنصر مشخص bind کنیم به طور با هر تغییر در این مقدار، عنصر مورد نظر نیز به روز می‌شود یا به بیانی دیگر همواره تغییرات observe خواهند شد. برای مثال:

```
Number of items : <span data-bind="text: myList().count"></span>
```

در نتیجه برای کار با KO وابستگی مستقیم به استفاده از JQuery وجود ندارد ولی این امکان هست که بتوانیم هم از JQuery و هم از KO در کنار هم به راحتی استفاده کنیم و از قدرت‌های هر دو فریم ورک بهره ببریم و البته KO جایگزینی برای JQuery نخواهد بود. در پست بعد، شروع به کار با KO آموزش داده خواهد شد. ادامه دارد...

نظرات خوانندگان

نویسنده: ابوالفضل رجب پور

تاریخ: ۱۳:۱۴ ۱۳۹۲/۰۶/۰۷

سلام و تشکر از آموزش خوبتون
 KNOCKOUT در مقایسه با angular ، کدام مناسبتر هستند؟
 آیا مقایسه این دو درست است؟
 شنیدم روی ویژوال استودیو 2013 مایکروسافت پیش فرض آنگولار رو استفاده کرده. این خودش خیلی نقطه قوت هست و حتما روش فکر کرده مایکروسافت!
 نظر شما چیه؟
 به طور کل برای spa چه مجموعه فریم ورکی رو پیشنهاد میدید؟
 مثلا ترکیب jquery + angular + requirejs چگونه؟

نویسنده: محسن خان

تاریخ: ۱۳:۴۲ ۱۳۹۲/۰۶/۰۷

در این مطلب فقط بوت استرپ نگارش 2 در MVC 5 پیش فرض شده. قبلا ناک آوت در MVC4 بود جزو اسکریپت‌های پیش فرض.

نویسنده: آرمان فرقانی

تاریخ: ۱۴:۱۷ ۱۳۹۲/۰۷/۰۶

این سوال خوبی است. اما گمان نمی‌کنم بشود پاسخ دقیقی به بخش کدام مناسبتر است به طور کلی داد. شاید بتوانید بر اساس علاقه به MVC یا MVVM یکی را برگزینید. برای کسانی هم که می‌خواهند یکی را شروع کنند شاید Knockout برای شروع با توجه به داکيومنت و بخش آموزش جالب آن بهتر باشد. همچنین مقایسه‌هایی مانند این یا بحث‌هایی مانند این کمک کننده است برای انتخاب بین این فریم ورک‌ها. البته هر دو فریم ورک مدرن و مناسب برای بسیاری موارد هستند. نظر شخصی من این است اگر ASP.NET MVC کار می‌کنید Angular را به صورت راه حل کلی دنبال کنید چون کمی کسب مهارت و آشنایی با تمام مفاهیم آن نسبت به Knockout بیشتر طول می‌کشد. و زمانی که صرف یافتن گزینه بهتر بین این دو می‌کنید را برای مطالعه Knockout با استفاده از مقالات همین سایت یا بخش آموزش سایت رسمی آن اختصاص دهید. گمان نمی‌کنم از صرف وقت برای این دو پشیمان شوید. هر کدام شیرینی خاص خود را دارند.