عنوان: KnockoutJs #5
نویسنده: مسعود پاکدل
تاریخ: ۱۳۹۲/۱۰/۱۵ ۸:۴۵
آدرس: www.dotnettips.info
گروه‌ها: Binding, KnockoutJS

Custom Binding در KO

در پست‌های قبلی(^ و ^ و ^) با انواع مقید سازی در KO آشنا شدید. اما در پیاده سازی، محدود به این نوع‌هایی ،click ،value text و ... نیستیم؛ بلکه می‌توانیم نوع مورد نظر برای عملیات مقید سازی را بنابر نیاز خود بسازیم که به آن‌ها Custom Binding گفته می‌شود. Custom Binding یکی از امکانات قدرتمند موجود در KO است و مورد اصلی استفاده آن در طراحی کامپوننت‌ها و ویجت‌ها می‌باشد.

مکانیزم پیاده سازی Custom Binding

برای شروع باید binding مورد نظر، به خاصیت ko.bindingHandlers رجیستر شود. سپس با تعیین کردن و شخصی سازی دو تابع init و update می‌توان نوع مقید سازی مورد نظر را تعریف کرد.

»init : این تابع فقط یک بار آن هم به ازای هر عنصری که عملیات مقید سازی را شامل می‌شود، فراخوانی خواهد شد.

»update : این تابع برای تعیین نوع عمل مورد انتظار در هنگام تغییر کردن مقدار عنصر DOM استفاده می‌شود.

برای مثال:

```
ko.bindingHandlers.myCustomBinding = {
    init: function(element, valueAccessor, allBindingsAccessor, viewModel , bindingContext) {

    },
    update: function(element, valueAccessor, allBindingsAccessor, viewModel, bindingContext) {

    }
};
```

پارامترهای توابع:

هر دو تابع بالا دقیقا دارای پنج پارامتر یکسان هستند که در زیر به تفصیل شرح داده شده‌اند:

»element : برای دسترسی مستقیم به عنصر DOMی که شامل مقید سازی است، می‌توان از این پارامتر استفاده کرد.

»valueAccessor : این پارامتر تابعی است که امکان دسترسی به هر آنچه را که به binding مورد نظر پاس داده باشیم، در اختیار ما قرار می‌دهد. برای مثال اگر observable را پاس داده باشیم، خروجی این تابع دقیقا همان observable خواهد بود. اگر از یک عبارت یا expression استفاده کرده باشیم خروجی این تابع برابر با حاصل آن عبارت خواهد بود.

»allBindingsAccessor : برای پیدا کردن لیست تمام عناصری است که به یک data-bind attribute مشترک اشاره می‌کنند.

»viewModel : برای دسترسی به viewModel عنصر مقید شده استفاده می‌شد. در knockout نسخه 3 به بعد این گزینه منسوخ شده است. به جای آن باید از پارامتر bindingContext.$data یا bindingContext.$rowData استفاده کرد.

»bindngContext : این پارامتر شی binding Context را که عنصر مورد نظر به آن مقید شده است، شامل می‌شود. این آبجکت شامل خواص $parent و $parents و $root است.

یک مثال ساده:

```
ko.bindingHandlers.jqButton= {
    init: function(element, valueAccessor) {
        var options = valueAccessor() || {};
        $(element).button(options);
    }
};
```

و روش استفاده از آن در عناصر DOM:

```
<button data-bind="click: greet, jqButton: { icons: { primary: 'ui-icon-gear' } }">Test</button>
```

دموی این مثال

استفاده از تابع update :

فرض کنید قصد داریم که با تغییر در مقدار یک متغیر، تغییرات مورد نظرمان در عنصر مقید شده نیز مشاهده شود. در این حالت باید از تابع update استفاده نمود. به مثال زیر دقت کنید:

```
ko.bindingHandlers.flash= {
    update: function(element, valueAccessor) {
        ko.utils.unwrapObservable(valueAccessor());
        $(element).hide().fadeIn(500);
    }
};
```

نکته : دستور ko.utils.unwrapObservable خاصیت مورد نظر را از حالت observe بودن خارج می‌کند.

دموی این مثال

ادامه دارد...

نظرات خوانندگان

نویسنده: موحدی نیا
تاریخ: ۱۳۹۳/۰۲/۰۴ ۱۰:۴۵

با سلام و تشکر از مطالب مفیدی که تو سایت قرار میدید
من یه پروژه case study رو چند روزی هست که شروع کردم و بدون مشکل کارم رو ادامه میدادم تااینکه به ویو ویرایش مشخصات افراد رسیدم.
سه تا از فیلدهای مربوط به افراد شامل کشور، استان و شهر میشه که تو View مربوط به افراد جدید این سه تا DropDownList با استفاده از Knockout پر میشن. بطوری که DropDownListهای مربوط به استان و شهر خالی هستند و با انتخاب کشور، استان پر میشه و با انتخاب استان، شهر پر میشه.
مشکل اینجاست که تو View ویرایش DropDownListهای استان و شهر در بارگذاری اولیه فرم پر نمیشن ولی با تغییر مقادیر کشور، استانها در DropDownList خودش پر میشه و این کار برای شهر هم به خوبی انجام میشه.
حالا میخام ببینم که چطور میشه این مشکل رو حل کرد

نویسنده: مسعود پاکدل
تاریخ: ۱۳۹۳/۰۲/۰۴ ۱۲:۴۸

برای اینکه بتوان پاسخ به سوال شما را بدون حدس گمان و به صورت قطعی بیان کنم لطفا کدهای مورد نظر را قرار بدید!

<< باگ را بدون باگ گزارش کن >>
<< آناتومی یک گزارش خطای خوب >>