

اغلب در حین Bind کردن Property ها در XAML به مشکل Bind نشدن بر می‌خوریم. من معمولا از روش زیر استفاده می‌کنم:

```
public class DatabindingDebugConverter : IValueConverter
{
    #region IValueConverter Members

    public object Convert(object value, Type targetType, object parameter,
        System.Globalization.CultureInfo culture)
    {
        Debugger.Break();
        return value;
    }

    public object ConvertBack(object value, Type targetType, object parameter,
        System.Globalization.CultureInfo culture)
    {
        Debugger.Break();
        return value;
    }

    #endregion IValueConverter Members
}
```

و در XAML :

```
<DataTemplate.Resources>
    <debug:DatabindingDebugConverter x:Key="databindingDebugConverter"/>
</DataTemplate.Resources>
<DataGrid ItemsSource="{Binding myViewModel, Converter={StaticResource databindingDebugConverter}}" />
```

و حال دو حالت می‌تواند اتفاق بیفتد :

1 - Break Point Hit نمی‌شود:

در این حالت مقدار myViewModel خالی (null) است و یا اصلا myViewModel در DataContext مربوط به DataGrid وجود ندارد در این صورت همچنین در پنجره Out Put Visual Studio:

```
System.Windows.Data Error: 35 : BindingExpression path error: 'X' property not found ...
```

و با search متن "System.Windows.Data Error: 35 : BindingExpression path error" در Out Put میتوان متوجه آن شد.

2 - Break Point Hit می‌شود:

در این حالت باید value را Watch کنیم (Shift+F9) تا ببینیم علت Bind نشدن چیست؟ شاید (در این مورد خاص) نوع myViewModel از IEnumerable نباشد ...

در حین بررسی و Debug ، شاید گاهی مسئله لاینحل به نظر برسد ، ولی به نظر من معمولا با کم و زیاد کردن آدرس (Binding Path) به یکی از دو حالت بالا خواهیم رسید ، مثلا زمانی که Path به صورت myViewModel.MyProperty.MyInnerProperty است ، باید Path را با حالات زیر توسط Converter مذکور تست کنیم:

```
Binding"{Path=myViewModel.MyProperty.MyInnerProperty , Converter="{StaticResource debugger}}}"
Binding"{Path=myViewModel.MyProperty, Converter="{StaticResource debugger}}}"
Binding"{Path=myViewModel, Converter="{StaticResource debugger}}}"
Binding"{Path=., Converter="{StaticResource debugger}}}"
```

امیدوارم از Binding تان لذت ببرید.

شاید تا به حال در یک برنامه سازمانی نیاز به Bind کردن یک Enum به کنترل‌های XAML به چشمتان خورده باشد ، روشی که من برای این کار استفاده می‌کنم توسط یک [Markup Extension](#) به صورت زیر است :

```
public class ByteEnumerationExtention : MarkupExtension
{
    public ByteEnumerationExtention(Type enumType)
    {
        this.enumType = enumType;
    }

    private Type enumType;

    public Type EnumType
    {
        get { return enumType; }
        private set
        {
            enumType = value;
        }
    }

    public override object ProvideValue(IServiceProvider serviceProvider)
    {
        return (from x in Enum.GetValues(EnumType).OfType<Enum>()
                select new EnumerationMember
                {
                    Value = GetValue(x),
                    Description = GetDescription(x)
                }).ToArray();
    }

    private byte? GetValue(object enumValue)
    {
        return Convert.ToByte(enumValue.GetType().GetField("value__").GetValue(enumValue));
    }

    public object GetObjectValue(object enumValue)
    {
        return enumValue.GetType().GetField("value__").GetValue(enumValue);
    }

    public string GetDescription(object enumValue)
    {
        var descAttrib = EnumType.GetField(enumValue.ToString())
            .GetCustomAttributes(typeof(DescriptionAttribute), false)
            .FirstOrDefault() as DescriptionAttribute;
        return descAttrib != null ? descAttrib.Description : enumValue.ToString();
    }
}

public class EnumerationMember
{
    public string Description { get; set; }

    public byte? Value { get; set; }
}
```

: XAML

```
<ComboBox ItemsSource="{Binding Source={ Extensions:ByteEnumerationExtention {x:Type type:MyEnum} }}"
DisplayMemberPath="Description"
SelectedValuePath="Value" SelectedValue="{Binding SelectedItemInViewModel}"/>
```

: Enum

```
public enum MyEnum : short
{
    [Description("گزینه 1")]
    Item1 = 1,

    [Description("گزینه 2")]
    Item2 = 2,

    [Description("گزینه 3")]
    Item3 = 3,

    [Description("گزینه 4")]
    Item4 = 4,

    [Description("گزینه 5")]
    Item5 = 5,

    .
    .
    .
}
```

: ViewModel در SelectedItem

```
short? selectedItemInViewModel;
public short? SelectedItemInViewModel
{
    get
    {
        return selectedItemInViewModel;
    }
    set
    {
        selectedItemInViewModel = value;
        RaisePropertyChanged("SelectedItemInViewModel");
        //do calculations if needed
    }
}
```

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 دقیقاً 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 دقیقاً 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 می‌شود. در knockout نسخه 3 به بعد این گزینه منسوخ شده است. به جای آن باید از پارامتر bindingContext.\$data یا bindingContext.\$rowData استفاده کرد.
 «bindingContext: 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 نظرمون 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:

۱۰:۴۵ ۱۳۹۳/۰۲/۰۴

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 تا اینکه 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 هستن 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:

۱۲:۴۸ ۱۳۹۳/۰۲/۰۴

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](#) >>