

پیاده سازی Extender

همان طور که در [پست‌های و مثال‌های قبلی](#) مشاهده شد با استفاده از Ko.Observable توانستیم عملیات مقید سازی را به کمک ویژگی‌های خواندن و نوشتن ساده، پیاده سازی نماییم. اما قصد داریم در طی عملیات نوشتن به جای یک tracking ساده تغییرات، بتوانیم یک سری عملیات مشخص را نیز اجرا نماییم. چیزی شبیه به AOP دنیای back-end. یعنی بتوانیم کد اصلی برنامه را در هنگام عملیات خواندن و نوشتن خاصیت‌ها، با یک سری کد مورد نظر مزین نماییم. برای این کار مفهوم extender در KO تعبیه شده است.

برای ساخت یک extender کافیت تابع مورد نظر را به عنوان آرگومان به شی ko.extenders پاس دهیم. پارامتر اول این تابع، شی observable شده مورد نظر و پارامتر دوم آن، شی option برای انجام یک سری تنظیمات یا فرستادن مقادیر مورد نظر به تابع است. خروجی این تابع نیز می‌تواند یک شی observable یا حتی یک شی [computed/observable](#) نیز باشد. یک مثال ساده برای extenderها به صورت زیر است:

```
ko.extenders.logOpt = function(target, option) {
    target.subscribe(function(newValue) {
        console.log(option + ": " + newValue);
    });
    return target;
};
```

در مثال بالا با ایجاد یک extender برای شی target که خود آن به عنوان آرگومان به تابع پاس داده می‌شود، به ازای هر تغییر در مقدار شی target، مقدار جدید نیز در console نمایش داده خواهد شد. مقدار چاپ شده در console برابر است با مقدار شی option + مقدار جدید شی target به ازای هر تغییر.

برای استفاده از این extender کافیت آن را در هنگام تعریف تابع observable برای خواص، به صورت زیر فراخوانی نمایید:

```
this.firstName = ko.observable("Masoud").extend({logOpt: "my first name"});
```

مقدار 'my first name' همان مقدار پاس داده شده در قالب شی option است. در نتیجه خروجی console به صورت زیر خواهد شد:

```
my first name : Masoud
```

پیاده سازی یک extender جهت اعلام هشدار برای مقادیر منفی

برای اینکه هنگام ورود داده‌ها توسط کاربر، بتوانیم با ورود مقادیر منفی یک هشدار (تغییر رنگ ورودی) اعلام کنیم، می‌توان به صورت زیر عمل نمود:

```
ko.extenders.negativeValueWarn = function (target, option) {
    target.hasWarning = ko.observable();

    function warn(newValue) {
        if(newValue && newValue.substring) {
            newValue = parseFloat(newValue);
        }
        target.hasWarning(newValue < 0 ? true : false);
    }

    warn(target());
    target.subscribe(warn);
};
```

```
return target;
};
```

تابع warn با در اختیار داشتن مقدار جدید و بررسی منفی یا مثبت بودن آن نتیجه را به تابع set شی hasWarning ارسال می‌کند.

یاد آوری : در KO برای انتساب مقدار جدید به خواصی که به صورت observable تعریف شده اند به صورت زیر:

```
target(NewValue) => فراخوانی به صورت تابع و پاس دادن مقدار جدید به آن
```

و برای به دست آوردن این مقادیر از اشیای Observable به صورت زیر عمل می‌نماییم:

```
target() => فراخوانی به صورت تابع بدون آرگومان
```

[خروجی مثال بالا](#)

پیاده سازی یک extender برای انتساب مقادیر Boolean به Radio Button ها

برای اینکه radio button ها نیز بتوانند فقط با مقادیر Boolean مقدار دهی شوند و از طرفی در هنگام عملیات مقید سازی و ارسال نتایج در قالب شی Json به سرور، بدون هیچ گونه تغییر و محاسبات مقادیر مورد نظر به صورت true/false (از نوع Boolean) باشند می‌توان به صورت زیر عمل نمود:

```
ko.extenders["booleanValue"] = function (target) {
    target.formattedValue = ko.computed({
        read: function () {
            if (target() === true) return "True";
            else if (target() === false) return "False";
        },
        write: function (newValue) {
            if (newValue) {
                if (newValue === "False") target(false);
                else if (newValue === "True") target(true);
            }
        }
    });
    target.formattedValue(target());
    return target;
};
```

در کد بالا یک sub-observable به نام formattedValue ایجا شده است و همان طور که ملاحظه می‌نمایید از نوع computed می‌باشد. در تابع read آن (هنگام عملیات مقید سازی برای خواندن مقادیر) اگر مقدار مورد نظر برابر با true از نوع boolean بود مقدار True (به صورت string) و اگر برابر با false بود مقدار False برگشت داده می‌شود. هنگام عملیات write بر عکس عمل خواهد شد.

با فرض اینکه کدهای Html صفحه به صورت زیر است:

```
<span>Do you want fries with that?</span>
<label>
    <input type="radio" name="question" value="True"
        data-bind="value: myValue.formattedValue" /> Yes
</label>
<label>
    <input type="radio" name="question" value="False"
        data-bind="value: myValue.formattedValue" /> No
</label>
```

Json Object مورد نظر که مقادیر boolean در آن به صورت true یا false است و بعد از عملیات مقید سازی در هنگام انتساب مقادیر، آن‌ها را تبدیل به True یا False برای المان‌های Html می‌کند. و در هنگام ورود اطلاعات توسط کاربر و انتساب آن‌ها به شی Json ، مقادیر تبدیل به true یا false از نوع boolean خواهند شد. برای استفاده از آن کافیست به صورت زیر عمل نمایید:

```
this.myValue= ko.observable(false).extend({ booleanValue: null });
```