

آموزش KnockoutJS - آموزش options binding در KnockoutJS

هدف

options binding کنترل می کند که چه option هایی باید در یک drop-down لیست (یعنی یک المان `<select>`) یا لیست چند انتخابی (multi-select list) (مثلا `<select size='6'>`) نمایش داده شود. این binding را تنها می توان کنار المان های `<select>` استفاده کرد.

مقداری که شما تخصیص می دهید باید یک آرایه یا یک آرایه ی قابل مشاهده باشد. در این صورت المان `<select>` به ازای هر یک از آیتم های موجود در آرایه ی شما، یک آیتم را نمایش می دهد.

نکته: برای اینکه مشخص کنید یا بخوانید که در یک لیست چند انتخابی کدام یک از option ها انتخاب شوند، از [selectedOptions binding](#) استفاده کنید. برای خواندن و نوشتن option انتخابی در یک لیست تک گزینه ای می توانید از [value binding](#) استفاده کنید.

مثال 1 : Drop-down لیست

```
<p>
  Destination country:
  <select data-bind="options: availableCountries"></select>
</p>

<script type="text/javascript">
  var viewModel = {
    // These are the initial options
    availableCountries: ko.observableArray(['France', 'Germany', 'Spain'])
  };

  // ... then later ...
  viewModel.availableCountries.push('China'); // Adds another option
</script>
```

مثال 2: لیست چند انتخابی

```
<p>
  Choose some countries you would like to visit:
  <select data-bind="options: availableCountries" size="5" multiple="true"></select>
</p>
```

```
<script type="text/javascript">
  var viewModel = {
    availableCountries: ko.observableArray(['France', 'Germany', 'Spain'])
  };
</script>
```

مثال 3 : Drop-down لیستی که نه تنها رشته ها را ارائه می کند، بلکه اشیاء قراردادی جاوا اسکریپت را نیز نمایش می دهد

```
<p>
  Your country:
  <select data-bind="options: availableCountries,
    optionsText: 'countryName',
    value: selectedCountry,
    optionsCaption: 'Choose...'"></select>
</p>
```

```
<div data-bind="visible: selectedCountry"> <!-- Appears when you select something -->
  You have chosen a country with population
  <span data-bind="text: selectedCountry() ? selectedCountry().countryPopulation :
'unknown'"></span>.
</div>
```

```
<script type="text/javascript">
  // Constructor for an object with two properties
  var Country = function(name, population) {
    this.countryName = name;
    this.countryPopulation = population;
  };

  var viewModel = {
    availableCountries : ko.observableArray([
      new Country("UK", 65000000),
      new Country("USA", 320000000),
      new Country("Sweden", 29000000)
    ]),
    selectedCountry : ko.observable() // Nothing selected by default
  };
</script>
```

مثال 4 : Drop-down لیستی که اشیاء قراردادی جاوااسکریپت را ارائه می دهد، که در آن text computed نمایش داده شده نقش تابع آیتم ارائه شده را ایفا می کند

```
<!-- Same as example 3, except the <select> box expressed as follows: -->
<select data-bind="options: availableCountries,
  optionsText: function(item) {
    return item.countryName + ' (pop: ' + item.countryPopulation + ' )'
  },
  value: selectedCountry,
  optionsCaption: 'Choose...'"></select>
```

دقت کنید که تنها تفاوت بین مثال های 3 و 4 مقدار optionsText است.

پارامترها

- پارامتر اصلی
پارامتر اصلی در اینجا یک آرایه یا یک آرایه ی قابل مشاهده است. KnockoutJS به ازای هر یک از آیتم ها یک <option> را به گره ی مربوطه ی <select> اضافه می کند. تمامی option های قبلی پاک خواهند شد.
در صورتی که مقدار پارامتر شما آرایه ای از رشته ها باشد، نیازی به استفاده از پارامترهای دیگر نیست. المان <select> به ازای هر یک از مقادیر رشته ای یک option را نمایش می دهد. با این حال اگر بخواهید این امکان را به کاربر بدهید تا از آرایه ای از اشیاء قراردادی جاوا اسکریپت (نه صرفا رشته ها) به انتخاب کردن بپردازد، در این صورت به پارامترهای optionsText و optionsValue که در زیر آمده اند مراجعه کنید.
اگر پارامتر شما مقدار قابل مشاهده ای باشد، در این صورت این binding ، option های موجود المان را هر زمان که مقدار آن ها تغییر کند، آپدیت می کند. در صورتی که این پارامتر قابل مشاهده نباشد، این binding تنها یک بار option های موجود المان را تنظیم می کند و دیگر آن ها را در آینده آپدیت نخواهد کرد.
- پارامترهای اضافی
 - optionsCaption
ممکن است که در برخی مواقع بخواهید که به صورت پیش فرض هیچ option مشخصی انتخاب نشده باشد، این در حالی است که معمولا drop-down لیست ها با یک یا چند آیتم انتخاب

شده آغاز می شوند. خب حالا چگونه می توانید از این بروز مشکل جلوگیری کنید؟ راه حل معمول استفاده از لیستی از option ها به صورت پیشوندی در کنار option ویژه ی زائد (special dummy option) است که کار این option ویژه خواندن "Select an item" یا "Please choose an option" یا جملاتی از این دست است. به گونه ای که این option ویژه انتخاب شده باشد.

انجام این کار راحت است: تنها کافیسیت یک پارامتر اضافی به نام optionsCaption را اضافه کنید. و مقدار آن را برای نمایش دادن رشته در نظر بگیرید. برای مثال :

```
<select data-bind='options: myOptions, optionsCaption: "Select an item...", value: myChosenValue'></select>
```

KnockoutJS لیست آیتم ها را به صورت پیشوندی در کنار لیستی قرار می دهد که متن "Select an item..." را نمایش می دهد و مقدار آن نامشخص است. بنابراین اگر myChosenValue مقدار نامشخص (undefined) را در خود نگهداری کند (که این کار را observable ها به صورت پیش فرض انجام می دهند)، در این صورت dummy option (همان option ویژه – مترجم –) انتخاب خواهد شد. در صورتی که پارامتر optionsCaption observable باشد، با تغییر مقدار observable متن آیتم اولیه آپدیت خواهد شد.

- optionsText

برای اینکه بدانید چگونه می توانید options را به یکی از آرایه های شیء قراردادی جاوا اسکریپت و نه فقط رشته ها مقید کنید، به مثال 3 مراجعه کنید. در این حالت شما باید انتخاب کنید که همزمان با نمایش متن در drop-down لیست یا لیست چند انتخابی کدام یک از مشخصه های اشیاء باید نمایش داده شود. در مثال 3 چگونگی مشخص کردن اسم این مشخصه با وارد کردن یک پارامتر اضافی به نام optionsText نشان داده شده است.

اگر نمی خواهید به ازای هر یک از آیتم های موجود در dropdown تنها به نمایش دادن مقدار مشخصه ی ساده به عنوان متن بسنده کنید، می توانید تابع جاوا اسکریپتی را به آپشن optionsText تحویل دهید و برای محاسبه ی متن نمایش داده شده منطق دلخواه خود را بر حسب شیء ارائه شده ایجاد کنید. به مثال 4 در بالا نگاه کنید. این مثال نشان می دهد که چگونه می توانید با قرار دادن مقادیر مشخصه های متعدد در کنار یکدیگر متن نمایش داده شده را تولید کنید.

- optionsValue

مانند پارامتر بالا برای اینکه تعیین کنید از کدام یک از مشخصه های اشیاء باید برای تنظیم value attribute موجود در المان های <option> تولید شده توسط KnockoutJS باید استفاده شود، می توانید از یک پارامتر اضافی به نام optionsValue استفاده کنید. همچنین برای اینکه بتوانید این value را تعیین کنید، از یک تابع جاوا اسکریپت استفاده کنید. این تابع

آیتم انتخاب شده را به عنوان تنها آرگومان خود دریافت کرده و یک مقدار رشته ای را برگشت می دهد. از این مقدار برگشتی می توان در value attribute مربوط به المان استفاده کرد. معمولا تنها کاربرد optionsValue تضمین این مطلب است که KnockoutJS می تواند بعد از اینکه شما مجموعه ای از option های موجود را آپدیت می کنید، گزینه های انتخاب شده را به درستی حفظ میکند. فرض کنید که به صورت مکرر از طریق فراخوانی های Ajax لیستی از اشیاء "car" را دریافت می کنید، برای اینکه مطمئن شوید ماشین انتخاب شده سر جای خود باقی می ماند باید optionsValue را بر روی "carId" یا هر شناسه ی ویژه ای که هر یک از اشیاء "car" دارند، قرار دهید. در این صورت KnockoutJS نمیتواند بفهمد که کدام یک از اشیاء "car" قبلی مطابق کدام یک از اشیاء "car" جدید است.

optionsIncludeDestroyed

گاهی اوقات ممکن است بخواهید یک آرایه ی ورودی را به عنوان deleted نشان کنید بدون اینکه سابقه ی موجودیت آن را کاملا از دست بدهید، به این کار پاک کردن غیرمخرب گفته می شود. برای اینکه بدانید چگونه باید این کار را انجام دهید به [destroy function](#) مراجعه کنید.

به صورت پیش فرض options binding از تمام آرایه های ورودی که به عنوان destroyed نشان شده اند عبور می کند (آن ها را مخفی میکند). برای اینکه ورودی های نابود شده را نمایش دهید، این پارامتر اضافی را به صورت زیر مشخص کنید:

```
<select data-bind='options: myOptions, optionsIncludeDestroyed: true'></select>
```

optionsAfterRender

اگر نیاز است که در المان های option تولید شده منطق های شخصی بیشتری را اجرا کنید، می توانید از optionsAfterRender callback استفاده کنید. به نکته ی 2 که در زیر آمده است مراجعه کنید.

selectedOptions

می توانید با استفاده از selectedOptions در یک لیست چند انتخابی حالت انتخاب را بخوانید و بنویسید. از نظر فنی این پارامتر binding مجزایی است و آموزش مخصوص به خود را دارد.

valueAllowUnset

اگر می خواهید KnockoutJS این امکان را برای مشخصه ی مدلتان ایجاد کند که مقادیری را دریافت کند که در المان <select> شما هیچ ورودی متناظری ندارد (و این مقادیر را با خالی کردن المان <select> نمایش دهد) در این صورت به آموزش مربوط به [valueAllowUnset](#) مراجعه کنید.

نکته 1 : در زمان تنظیم کردن یا تغییر دادن option ها، گزینه های انتخاب شده حفظ می شوند.

زمانی که options binding مجموعه ای از option های موجود در المان <select> شما را تغییر می دهد، KnockoutJS تا حد امکان انتخاب کاربر را دست نخورده باقی می گذارد. به همین دلیل در drop-down لیست های تک انتخابی مقدار option انتخاب شده ی قبلی همچنان انتخاب شده باقی می ماند و در لیست های چند انتخابی تمامی مقادیر option انتخاب شده ی قبلی نیز همچنان انتخاب شده باقی می ماند (البته مگر اینکه یک یا بیشتر از یک عدد از این انتخاب ها را پاک کرده باشید).

به همین دلیل است که options binding سعی می کند که مستقل از value binding (که وظیفه ی کنترل کردن انتخاب ها را در لیست های تک انتخابی برعهده دارد) و مستقل از selectedOptions (که وظیفه ی کنترل کردن انتخاب ها در لیست های چند انتخابی را برعهده دارد) باشد.

آموزش KnockoutJS – آموزش پسپردازش option های تولید شده

اگر نیاز است که در المان های option تولید شده منطق های شخصی بیشتری را اجرا کنید می توانید از optionsAfterRender callback استفاده کنید. این تابع callback هر زمان که المان option ای در لیست وارد شود، همراه با پارامترهای زیر احضار می شود:

1. المان option وارد شده

2. data item در برابر data item که مقید است، یا undefined برای المان caption

در مثال زیر برای اضافه کردن یک disable binding به هر یک از option ها از optionsAfterRender استفاده شده است.

```
<select size=3 data-bind="
  options: myItems,
  optionsText: 'name',
  optionsValue: 'id',
  optionsAfterRender: setOptionDisable">
</select>
```

```
<script type="text/javascript">
var vm = {
  myItems: [
    { name: 'Item 1', id: 1, disable: ko.observable(false)},
    { name: 'Item 3', id: 3, disable: ko.observable(true)},
    { name: 'Item 4', id: 4, disable: ko.observable(false)}
  ],
  setOptionDisable: function(option, item) {
    ko.applyBindingsToNode(option, {disable: item.disable}, item);
  }
};
```

```
ko.applyBindings(vm);  
</script>
```

وابستگی ها

به غیر از کتابخانه ی اصلی KnockoutJS وابستگی دیگری ندارد.