

آموزش KnockoutJS checked binding - آموزش KnockoutJS در checked binding

هدف

checked binding یک checkable form control (همان checkbox) (<input type='checkbox'>) یا یک radio button (<input type='radio'>) را به یک مشخصه ی موجود در view model شما متصل می کند. زمانی که کاربر تیک form control مربوطه را می زند، مقدار view model شما آپدیت می شود. همچنین وقتی شما مقدار view model تان را آپدیت کنید، تیک form control موجود در صفحه گذاشته با برداشته می شود. نکته: در text box ها drop-down لیست ها و همه ی form control های غیرقابل تیک زدن برای خواندن و نوشتن مقدار المان از [value binding](#) استفاده کنید، نه از checked binding.

مثال با checkbox

```
<p>Send me spam: <input type="checkbox" data-bind="checked: wantsSpam" /></p>
```

```
<script type="text/javascript">
  var viewModel = {
    wantsSpam: ko.observable(true) // Initially checked
  };

  // ... then later ...
  viewModel.wantsSpam(false); // The checkbox becomes unchecked
</script>
```

مثال اضافه کردن قید checkbox ها به یک آرایه

```
<p>Send me spam: <input type="checkbox" data-bind="checked: wantsSpam" /></p>
<div data-bind="visible: wantsSpam">
  Preferred flavors of spam:
  <div><input type="checkbox" value="cherry" data-bind="checked: spamFlavors" />
  Cherry</div>
  <div><input type="checkbox" value="almond" data-bind="checked: spamFlavors" />
  Almond</div>
  <div><input type="checkbox" value="msg" data-bind="checked: spamFlavors" />
  Monosodium Glutamate</div>
```

```
</div>
```

```
<script type="text/javascript">
```

```
  var viewModel = {
```

```
    wantsSpam: ko.observable(true),
```

```
    spamFlavors: ko.observableArray(["cherry", "almond"]) // Initially checks the Cherry and  
Almond checkboxes
```

```
  };
```

```
  // ... then later ...
```

```
  viewModel.spamFlavors.push("msg"); // Now additionally checks the Monosodium  
Glutamate checkbox
```

```
</script>
```

مثال اضافه کردن radio button ها

```
<p>Send me spam: <input type="checkbox" data-bind="checked: wantsSpam" /></p>
```

```
<div data-bind="visible: wantsSpam">
```

```
  Preferred flavor of spam:
```

```
  <div><input type="radio" name="flavorGroup" value="cherry" data-bind="checked:  
spamFlavor" /> Cherry</div>
```

```
  <div><input type="radio" name="flavorGroup" value="almond" data-bind="checked:  
spamFlavor" /> Almond</div>
```

```
  <div><input type="radio" name="flavorGroup" value="msg" data-bind="checked:  
spamFlavor" /> Monosodium Glutamate</div>
```

```
</div>
```

```
<script type="text/javascript">
```

```
  var viewModel = {
```

```
    wantsSpam: ko.observable(true),
```

```
    spamFlavor: ko.observable("almond") // Initially selects only the Almond radio button
```

```
  };
```

```
  // ... then later ...
```

```
  viewModel.spamFlavor("msg"); // Now only Monosodium Glutamate is checked
```

```
</script>
```

پارامترها

- پارامتر اصلی
 KnockoutJS حالت چک شده ی (checked) المان را برای مطابقت با مقدار پارامتر شما تنظیم می کند. تمامی حالت های چک شده ی قبلی overwrite خواهند شد. اینکه چگونه پارامتر شما تفسیر شود به نوع المانی بستگی دارد که شما می خواهید این پارامتر را به آن مقید کنید:
- در checkbox ها زمانی که مقدار پارامتر برابر با true باشد، KnockoutJS المان را بر روی *checked* قرار می دهد. و اگر این مقدار برابر با false باشد، آن را بر روی *unchecked* قرار می دهد. در صورتی که شما مقداری را وارد کنید که واقعا بولی نباشد، این مقدار با بی دقتی تفسیر می شود. این یعنی تمامی اعداد غیرصفر و اشیاء غیر تهی و رشته های غیرخالی تماما به عنوان true تفسیر می شوند. همچنین صفر، تهی، نامشخص و رشته های خالی به عنوان false تفسیر می شوند.
 زمانی که کاربر تیک checkbox را می زند یا آن را بر می دارد، در این صورت KnockoutJS به ترتیب مشخصه ی مدل شما را بر روی true یا false قرار می دهد.
 در صورتی که پارامتر شما به یک آرایه تبدیل شود، ملاحظات ویژه ای نیاز خواهد بود. KnockoutJS در این حالت در صورتی که مقدار با یکی از آیتم های موجود در آرایه مطابقت داشته باشد، المان را بر روی *checked* قرار می دهد و در صورتی که این مقدار در آرایه وجود نداشته باشد، آن را بر روی *unchecked* قرار می دهد.
 زمانی که کاربر تیک checkbox را می زند یا آن را برمی دارد، در این صورت به ترتیب KnockoutJS مقدار را به آرایه اضافه می کند یا آن را از آرایه حذف می کند.
- در radio button ها اگر و تنها اگر مقدار پارامتر با مقدار value attribute گره ی radio button یا با مقدار مشخص شده توسط پارامتر checkedValue برابر باشد، المان را بر روی حالت *checked* قرار می دهد. در مثال قبل تنها زمانی که مشخصه ی spamFlavor مربوط به view model برابر با "almond" شد، radio button با مقدار value="almond" تیک خورد.
 زمانی که کاربر radio button دیگری را انتخاب کند، در این صورت KnockoutJS مشخصه ی مدل شما را به گونه ای تنظیم خواهد کرد که با مقدار radio button انتخاب شده برابر شود. در مثال قبل کلیک کردن بر روی radio button ای که مقدار آن برابر با value="cherry" بود، باعث شد viewModel.spamFlavor بر روی "cherry" قرارگیرد.
 البته از حالت بالا زمانی می توانید بیشترین استفاده را بکنید که المان های متعددی از radio button ها داشته باشید. به گونه ای که این المان ها به تنها یک مشخصه ی مدل مقید باشند. برای اینکه مطمئن شوید تنها یکی از این radio button ها در هر زمانی می تواند تیک زده شود، می توانید تمام name attribute ها را بر روی یک arbitrary common value قرار دهید (مثلا مقدار flavorGroup موجود در مثال قبل). با انجام این کار تمام این radio button ها را در گروهی قرار می دهید که تنها یکی از آن ها می تواند انتخاب شود.
 در صورتی که پارامتر شما مقدار قابل مشاهده ای باشد، binding هر زمان که این مقدار تغییر کند حالت checked المان را آپدیت می کند. اما در صورتی که این پارامتر قابل مشاهده نباشد، تنها یک بار حالت checked المان را تنظیم کرده و دیگر در آینده آن را آپدیت نخواهد کرد.
- پارامتر اضافی

▪ checkedValue

اگر شما دارای `checkedValue` هم باشد، این پارامتر با استفاده از `checked` binding به جای `value attribute` المان، مقدار را تعریف می کند. این پارامتر در صورتی به کمکتان می آید که بخواهید این مقدار چیزی غیر از یک رشته (مانند عدد صحیح یا شیء) باشد. یا اینکه بخواهید `value` به صورت پویا تنظیم شود. در مثال زیر، خود `item object` ها (نه رشته های `itemName` آن ها) زمانی که `checkbox` های متناظر آن ها تیک زده شوند، در `chosenItems` لحاظ می شوند:

```
<!-- ko foreach: items -->
  <input type="checkbox" data-bind="checkedValue: $data, checked: $root.chosenItems" />
  <span data-bind="text: itemName"></span>
<!-- /ko -->
```

```
<script type="text/javascript">
  var viewModel = {
    items: ko.observableArray([
      { itemName: 'Choice 1' },
      { itemName: 'Choice 2' }
    ]),
    chosenItems: ko.observableArray()
  };
</script>
```

اگر پارامتر `checkedValue` شما مقدار قابل مشاهده ای باشد، هر زمان که این مقدار تغییر کند و این المان در حال حاضر تیک خورده باشد، در این صورت `binding` مشخصه ی مدل `checked` را آپدیت می کند. در `checkbox` ها این المان مقدار قدیمی را از آرایه پاک کرده و مقدار جدید را اضافه می کند. این پارامتر در `radio button` ها تنها مقدار مدل را آپدیت می کند.

وابستگی ها

به غیر از کتابخانه ی اصلی KnockoutJS وابستگی دیگری ندارد.