Jonathan Disla

Discussion 7.1

Web 330

Prof. Krasso

Knockout JS comes with some very useful pre-defined bindings such as form field, component, display, and template bindings. However, there are situations where custom bindings are necessary if they need to be reusable. Custom bindings have to be added to the bindingHandlers property. A custom binding can be made up of two properties which are init and update which both take in a function. The init function will only execute once and it is used to initialize event handlers and set up the initial state. The update function executes every time the observable changes. The custom bindings need parameters such as element, value accessor, all bindings, view model, and binding context.

Firstly we provide the DOM element to be targetted using a value accessor which gets the current model property involved in the binding. To access all the model values bound in the element, all bindings are used in this case you can use this function to check if the element has a specific binding or to get the binding. Finally, the binding context holds the binding context available to the element's binding.

I will provide an example of custom bindings in this case let's say that we want to update a form field:

<html>

<head>

<script type='text/javascript' src='knockout-3.5.1.js'></script>

</head>

<body>

<div **id**="main-wrapper">

<!-- *ORIGINAL* -->

<!-- *<input id= "nameInput" data-bind="value: userName" placeholder="Enter Name"/><button type="button" class="btn btn-primary" id="submitName">Submit</button>* -->

<!-- *Custom Binding* -->

<input **id**= "nameInput" data-bind="display:inactive" placeholder="Enter Name"/><button type="button" class="btn btn-primary" **id**="submitName">Submit</button>

</div>

<script>

ko**.**bindingHandlers**.**Display ={

**update**: **function**(element,valueAccessor, allBindings, viewModel, bindingContext){

**let** value = ko**.**utils**.unwrapObserbable**(**valueAccessor**());

**let** status = value;

element**.**display = status;

}

}

**function** **ViewModel**(){

**var** self = **this**;

**this.**name = name;

**let** value;

//*Replacing this with an custom binding*

// *self.userName= ko.observable("");*

self**.**inactive = ko**.observable**(value)

}

ko**.applyBindings**(**new** **ViewModel**(),

document**.querySelector**("#main-wrapper"))

</script>

</body>

</html>

Reference List

abouscript. “KnockoutJS Sliders, DOM Properties, Custom Bindings.” YouTube. 25 July, 2013.

https://www.youtube.com/watch?v=siif1lEfM7w&ab\_channel=aboutscript

BestDotNetTraining. “Part 12- What is Custom Binding in Knockout JS.” YouTube 30 March 2016.

<https://www.youtube.com/watch?v=qQxD1DNAOPc&ab_channel=BestDotNetTraining>