

Custom Binding Handlers and Persisting Data

John Papa

[@john_papa](https://twitter.com/john_papa)

<http://johnpapa.net>



Outline

- Custom Binding Handlers
- Unobtrusive JavaScript
- Saving/Loading Data
- Change Tracking
- Mapping

What Are Custom Binding Handlers?

- **Just another binding**
 - Can't find the right built-in binding? Make your own!
- **Examples**
 - Animate a transition for an item
 - fadeVisible
 - Integrate with jQuery UI
 - jqButton
 - Get creative
 - starRating

init and update

Runs the first time
the binding is eval'd

```
ko.bindingHandlers.fadeVisible = {  
  init: function(element, valueAccessor) {  
    // ...  
  },  
  
  update: function(element, valueAccessor) {  
    // ...  
  }  
}
```

After init, and every time
one of its observables
changes

Parameters

The bound
element in the
DOM

What is
passed to
the binding

All other bindings in the
same data-bind attribute

```
ko.bindingHandlers.fadeVisible = {  
  init: function(element, valueAccessor, allBindingsAccessor, viewModel) {  
    var shouldDisplay = valueAccessor();  
    $(element).toggle(shouldDisplay);  
  },  
  
  update: function(element, valueAccessor, allBindingsAccessor, viewModel) {  
    var shouldDisplay = valueAccessor();  
    shouldDisplay ? $(element).fadeIn() : $(element).fadeOut();  
  }  
}
```

The overall view
model

DEMO

Custom Binding Handlers



Outline

- Custom Binding Handlers
- **Unobtrusive JavaScript**
- Saving/Loading Data
- Change Tracking
- Mapping

Unobtrusive Event Handlers and Helpers

```
<div class="closeIcon"  
  data-bind="click:$parent.removeProduct"></div>
```

```
<div class="closeIcon"></div>
```

```
//In the view model
```

```
removeProduct = function (p) {  
  products.remove(p);  
}
```

*data available for
binding on the element*

```
var itemSelector = "ul li";  
$(itemSelector).on("click", "div.closeIcon",  
  function () {  
    my.vm.removeProduct(ko.dataFor(this));  
  });
```


Unobtrusive JavaScript: Binding Mappings

Move *data-bind*
to JavaScript/jQuery



```
$("#total").attr("data-bind", "value: grandTotal");  
  
$("#total").attr("data-bind", "css: {hilite: metGoal}");
```

DEMO

Unobtrusive JavaScript Events



Outline

- Custom Binding Handlers
- Unobtrusive JavaScript
- Saving/Loading Data
- Change Tracking
- Mapping

Loading / Saving Data

- **Add Controller**
 - Simple load and save methods
- **Add Models**
- **Add**
 - `papa.ajaxservice.js`
 - `dataservice.js`

DEMO

Saving / Loading Data



Outline

- Custom Binding Handlers
- Unobtrusive JavaScript
- Saving/Loading Data
- **Change Tracking**
- Mapping

Change Tracking

```
vm.tracker = new ChangeTracker(viewModel);  
  
// Did It Change?  
vm.tracker().somethingHasChanged();  
  
// Hook this into your view model functions  
// (ex: load, save) ...  
// Resync Changes  
vm.tracker().markCurrentStateAsClean;
```

DEMO

Change Tracking



Outline

- Custom Binding Handlers
- Unobtrusive JavaScript
- Saving/Loading Data
- Change Tracking
- Mapping

Mapping JSON to Objects

- **Creating observables can be monotonous**
- **Especially ...**
 - Ajax to fetch data and return JSON
- **Knockout JS Mapper**
 - Community created plug in
 - Map
 - Converts JSON to an object with observables
 - Unmap
 - Converts mapped object back to a regular JavaScript object

Mapping

```
load = function () {  
    my.PersonSvc.getPersons(  
        function (result) {  
            model(ko.mapping.fromJS(result));  
        }  
    );  
};
```

```
save = function () {  
    var person = ko.mapping.toJSON(model());  
    my.PersonSvc.savePerson(person,  
        function (result) {  
            // ...  
        }  
    );  
};
```

Summary

- Custom Binding Handlers
- Unobtrusive JavaScript
- Saving/Loading Data
- Change Tracking
- Mapping

For more in-depth **online** developer **training** visit



on-demand content from authors you **trust**