

# Framework

Keith Burnell

@keburnell



# Agenda

- DataSource
- MVVM
- Templates
- Validator

# **DataSource**

- **Abstraction for client-side data**
- **Local and remote data**
- **CRUD**
- **Sorting, Paging, Grouping, Aggregates**

# Local Data

```
<script type="text/javascript">
    $(document).ready(function () {
        var cars = [
            { year: 2010, make: "Ferrari", model: "California" },
            { year: 2012, make: "Ferrari", model: "458 Italia" },
            { year: 2012, make: "Lamborghini", model: "Aventador" },
            { year: 2004, make: "Lamborghini", model: "Gallardo" },
            { year: 2006, make: "Porsche", model: "997 GT3 RS" },
            { year: 1998, make: "McLaren", model: "F1" },
            { year: 2010, make: "Bugatti", model: "Veyron" },
            { year: 2006, make: "Koenigsegg", model: "CCX" }
        ];
        var carsDataSource = new kendo.data.DataSource({ data: cars });
    });
</script>
```

# Remote Data

```
<script type="text/javascript">
    $(document).ready(function () {
        var twitterSearch = new kendo.data.DataSource({
            transport: {
                read: {
                    url: "http://search.twitter.com/search.json",
                    dataType: "jsonp",
                    data: { q: "#kendoui" }
                }
            },
            schema: { data: "results" }
        });
    });
</script>
```

# OData

```
<script type="text/javascript">
  $(document).ready(function () {
    var movies = new kendo.data.DataSource({
      type: "odata",
      transport: {
        read: "http://odata.netflix.com/Catalog/Titles"
      }
    });
  });
</script>
```

# MVVM Framework

- Declarative bindings
- Automatic two-way data synchronization
- Dependent properties
- Fast

# ViewModel

- Representation of your data (Model)
- Displayed via the View
- Declared using the `kendo.observable` function

```
var viewModel = kendo.observable({  
  make: "Ferrari",  
  model: "California",  
  messageIsVisible: function () {  
    return this.get("make").length != 0;  
  }  
});
```



# Templates

- Reusable chunks of HTML
- Can be automatically merged with JavaScript data
- Emphasis on performance
- Hash (#) syntax
  - Render literal values:                   #=   #
  - Render HTML-encoded values:       #:   #
  - Render JavaScript code:           #    #

```
<script type="text/x-kendo-template" id="template">
  <h1>#= listName #</h1>
  <ul>
    # for (var i = 0; i < cars.length; i++) { #
      <li>#= cars[i]["make"] # - #= cars[i]["model"] #</li>
    # } #
  </ul>
</script>
```

# Validator

- Client-side form validation
- Built on HTML5 form validation
- Built in rules
- Custom rules

# Summary

- DataSource provides an abstraction
- MVVM gives us automatic two-way data synchronization
- Templating for reusable chunks of HTML
- Validator for client-side validation