

Strongly-Typed Binding



Matt Honeycutt

@matthoneycutt | <http://trycatchfail.com/strongly-typed-ajs>

Before

```
<tr>
  <th>E-mail</th>
  <td>
    Work: {{vm.customer.WorkEmail}}<br />
    Home: {{vm.customer.HomeEmail}}
  </td>
  <th>Phone</th>
  <td>
    Office: {{vm.customer.WorkPhone}}<br />
    Mobile: {{vm.customer.HomePhone}}
  </td>
</tr>
<tr>
  <th>Work Address</th>
  <td style="white-space: pre">{{vm.customer.WorkAddress}}</td>
  <th>Home Address</th>
  <td style="white-space: pre">{{vm.customer.HomeAddress}}</td>
</tr>
```

After

```
<tr>
  <th>E-mail</th>
  <td>
    Work: @customer.BindingFor(x => x.WorkEmail)<br />
    Home: @customer.BindingFor(x => x.HomeEmail)
  </td>
  <th>Phone</th>
  <td>
    Office: @customer.BindingFor(x => x.WorkPhone)<br />
    Mobile: @customer.BindingFor(x => x.HomePhone)
  </td>
</tr>
<tr>
  <th>Work Address</th>
  <td style="white-space: pre">@customer.BindingFor(x => x.WorkAddress)</td>
  <th>Home Address</th>
  <td style="white-space: pre">@customer.BindingFor(x => x.HomeAddress)</td>
</tr>
```

Overview



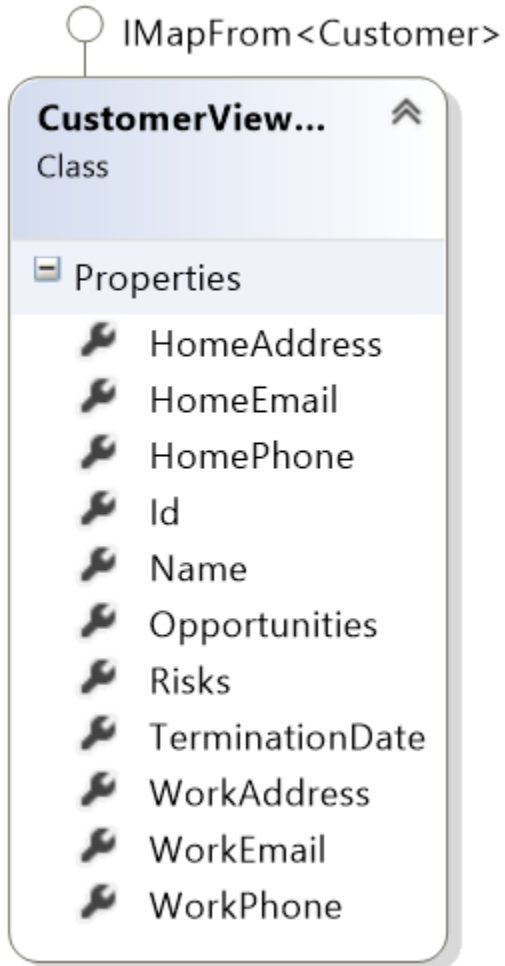
Angular expression helper

Binding helper

Extending JavaScript

ngRepeat helper

From C# Models to AngularJS Expressions



C# lambda

`x => x.Name`



Client-side expression

`vm.customer.name`

Controller-as syntax

```
<div class="panel panel-default"
  ng-controller="CustomerController as vm">
  <div class="panel-heading">
    {{vm.customer.name}} <a href="" ng-click="vm.edit()">
```

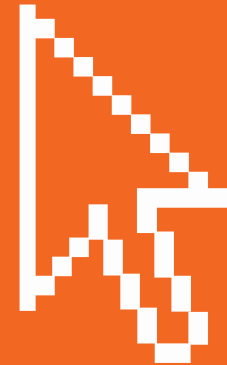
From C# to AngularJS

```
<tr class="info text-center" ng-show="vm.customer.TerminationDate">  
  <td colspan="4">  
    Inactive Customer  
  </td>  
</tr>
```



```
<tr class="info text-center" ng-show="@customer.ExpressionFor(x => x.TerminationDate)">  
  <td colspan="4">  
    Inactive Customer  
  </td>  
</tr>
```

Converting Lambda Expressions from C# to AngularJS



From C# Lambdas to AngularJS Bindings



ExpressionFor helper

```
<tr class="info text-center" ng-show="@customer.ExpressionFor(x => x.TerminationDate)">
  <td colspan="4">
    Inactive Customer
  </td>
</tr>
```


From C# Lambdas to AngularJS Bindings

```
<th>E-mail</th>
<td>
  Work: {{vm.customer.WorkEmail}}<br />
  Home: {{vm.customer.HomeEmail}}
</td>
```

E-mail

Work: zack@work.com
Home: zack@home.com

```
<th>E-mail</th>
<td>
  Work: {{vm.customer.workEmail}}<br />
  Home: {{vm.customer.HomeEmail}}
</td>
```



E-mail

Work: _____?
Home: zack@home.com

From C# Lambdas to AngularJS Bindings

```
<th>E-mail</th>
<td>
  Work: {{vm.customer.WorkEmail}}<br />
  Home: {{vm.customer.HomeEmail}}
</td>
```

(Not) good

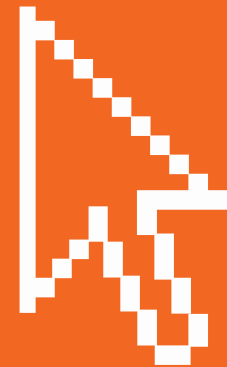
```
<th>E-mail</th>
<td>
  Work: {{@customer.ExpressionFor(x => x.WorkEmail)}}<br />
  Home: {{@customer.ExpressionFor(x => x.HomeEmail)}}
</td>
```

Better

```
<th>E-mail</th>
<td>
  Work: @customer.BindingFor(x => x.WorkEmail)<br />
  Home: @customer.BindingFor(x => x.HomeEmail)
</td>
```

Best!

Strongly-typed AngularJS Bindings



The Problem with Length

```
<p ng-show="vm.customer.Opportunities.length == 0">
  There are no opportunities for this customer.
</p>
<div ng-repeat="opportunity in vm.customer.Opportunities">
  <hr ng-hide="$index == 0" />
  <h3>
    {{opportunity.Title}}
  </h3>
  <p>{{opportunity.Description}}</p>
</div>
```

The Problem with Length



```
<p ng-show="@customer.ExpressionFor(x => x.Opportunities.Count) == 0">  
    There are no opportunities for this customer.  
</p>  
<div ng-repeat="opportunity in vm.customer.Opportunities">  
    <hr ng-hide="$index == 0" />  
    <h3>  
        {{opportunity.Title}}  
    </h3>  
    <p>{{opportunity.Description}}</p>  
</div>
```

The Problem with Length



undefined!

```
<p ng-show="vm.customer.opportunities.count == 0">
  There are no opportunities for this customer.
</p>
<div ng-repeat="opportunity in vm.customer.Opportunities">
  <hr ng-hide="$index == 0" />
  <h3>
    {{opportunity.Title}}
  </h3>
  <p>{{opportunity.Description}}</p>
</div>
```

The Problem with Length

Option 1....

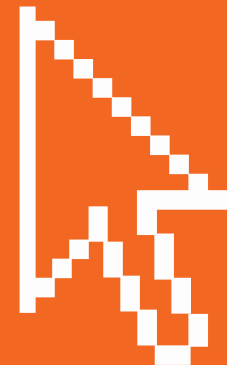


Option 2....

```
Object.defineProperty(Array.prototype, 'count', {  
  get: function () { return this.length; }  
});
```



Implementing Array.count



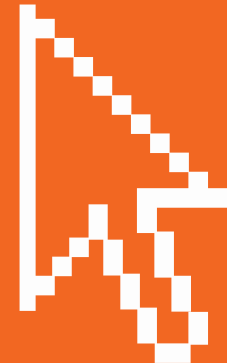
The ngRepeat Directive

```
<div ng-repeat="opportunity in vm.customer.Opportunities">
  <hr ng-hide="$index == 0" />
  <h3>
    {{opportunity.Title}}
  </h3>
  <p>{{opportunity.Description}}</p>
</div>
```

The ngRepeat Directive

```
@using (var opportunity = customer.Repeat(x => x.Opportunities, "opportunity"))
{
    <hr ng-hide="$index == 0" />
    <h3>
        @opportunity.BindingFor(x => x.Title)
    </h3>
    <p>@opportunity.BindingFor(x => x.Description)</p>
}
```

Strongly-typed ngRepeat



What We've Accomplished

Before:

```
<tr class="info text-center" ng-show="vm.customer.TerminationDate">
  <td colspan="4">
    Inactive Customer
  </td>
</tr>
```

After:

```
<tr class="info text-center" ng-show="@customer.ExpressionFor(x => x.TerminationDate)">
  <td colspan="4">
    Inactive Customer
  </td>
</tr>
```

What We've Accomplished

Before:

```
<th>E-mail</th>
<td>
    Work: {{vm.customer.WorkEmail}}<br />
    Home: {{vm.customer.HomeEmail}}
</td>
```

After:

```
<th>E-mail</th>
<td>
    Work: @customer.BindingFor(x => x.WorkEmail)<br />
    Home: @customer.BindingFor(x => x.HomeEmail)
</td>
```

What We've Accomplished

Before:



```
<p ng-show="vm.customer.Opportunities.length == 0">  
  There are no opportunities for this customer.  
</p>
```

After:

```
<p ng-show="@customer.ExpressionFor(x => x.Opportunities.Count) == 0">  
  There are no opportunities for this customer.  
</p>
```

What We've Accomplished

Before:

```
<div ng-repeat="opportunity in vm.customer.Opportunities">
  <hr ng-hide="$index == 0" />
  <h3>
    {{opportunity.Title}}
  </h3>
  <p>{{opportunity.Description}}</p>
</div>
```

After:

```
@using (var opportunity = customer.Repeat(x => x.Opportunities, "opportunity"))
{
  <hr ng-hide="$index == 0" />
  <h3>
    @opportunity.BindingFor(x => x.Title)
  </h3>
  <p>@opportunity.BindingFor(x => x.Description)</p>
}
```

Up Next...

Building strongly-typed
forms!

