

insum

Oracle Specialists
APEX Experts

Deliver a Knockout with
your APEX Applications

Jorge Rimblas



inum.ca

Jorge Rimblas



Senior APEX Consultant

 @rimblas
rimblas.com/blog

- Contributor to "Expert Oracle Application Express, 2nd Edition" with "Themes & Templates" chapter
- Oracle DB since 1995
- APEX since it was HTMLDB in 2004
- Always involved in web technologies
- jrimblas in OTN Forums



Quick Facts



Age: 14 years!

Staff: 80+ employees

60 consultants/developers

APEX Solutions: 12 Years!

Largest APEX practice in North America

Oracle Center of Excellence



Knockout.

Agenda

- Why?
- What is KnockoutJS?
- Data Binding
- Two-Way Data Binding
- Using with APEX
- More APEX examples

Why?

Knockout.

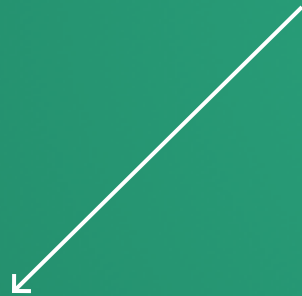
Solid Track Record

Used by

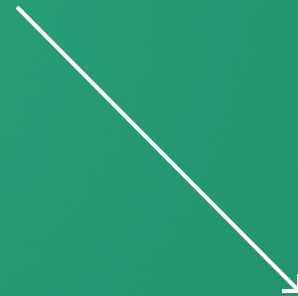


Library

Not a framework



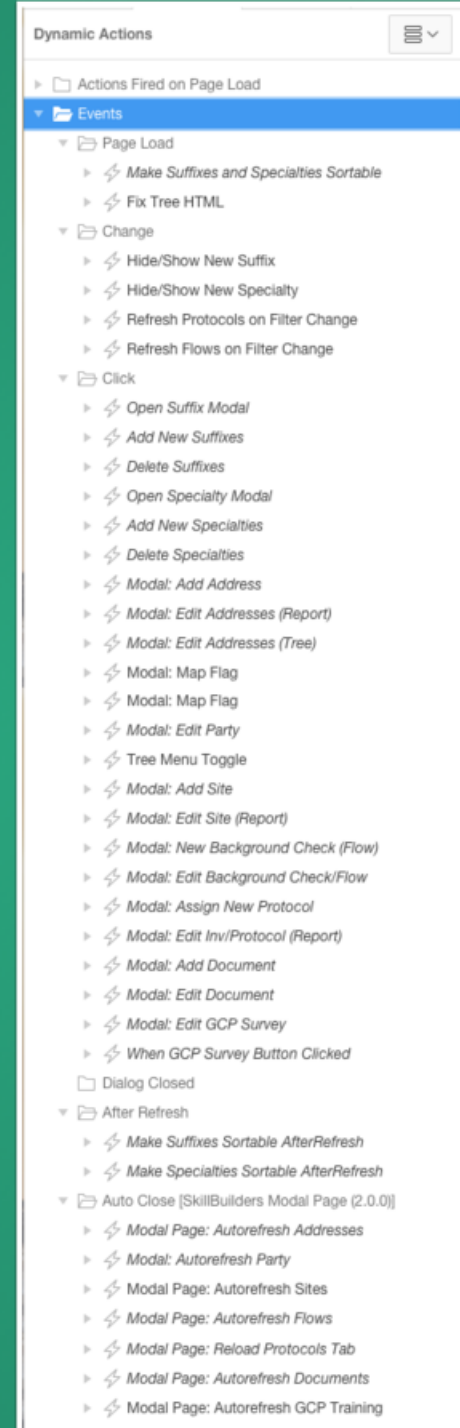
jQuery is a library



Good news for APEX

More dynamic than
Dynamic Actions

Dynamic Actions!



What is
Knockout.

?

“ Knockout is a JavaScript library that helps you to create rich, responsive display and editor user interfaces with a clean underlying data model.

** Source knockoutjs.com*



Declarative Bindings



Automatic UI Refresh



Dependency Tracking



Templating



Declarative Bindings



```
select * from emp where empno = :empno
```

```
<span data-bind="text: empno"></span>
```



```
<span data-bind="text: empno"></span>
```

JavaScript ViewModel contains data

```
function emp() {  
    this.empno = 7839;  
}
```



```
<span data-bind="text: empno"></span>  
<span data-bind="text: ename"></span>  
<span data-bind="text: job"></span>
```

JavaScript ViewModel contains data

```
function emp() {  
    this.empno = 7839;  
    this.ename = 'KING';  
    this.job    = 'PRESIDENT';  
    this.sal    = 5000;  
}
```

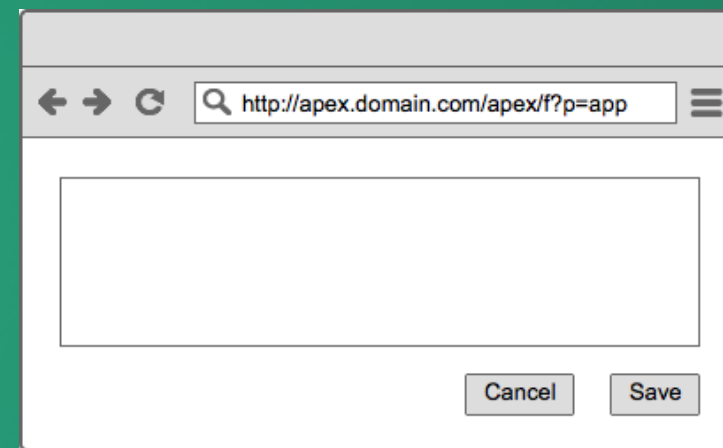


```
function emp( ) {  
    this.empno = 7839;  
    this.ename = 'KING';  
    this.job    = 'PRESIDENT';  
    this.sal    = 5000;  
}  
  
var viewModel = new emp();  
  
ko.applyBindings(viewModel);
```



Automatic UI Refresh

```
...JavaScript...  
ko.applyBindings( );
```





Two-Way Data Bind



observables

```
function empModel(data) {  
  this.empno = data.empno;  
  this.ename = ko.observable(data.ename);  
  this.job   = ko.observable(data.job);  
  this.sal   = ko.observable(data.sal);  
}
```



observables

```
ko.observable()  
ko.observableArray()
```



Dependency Tracking



Dependency Tracking

computed

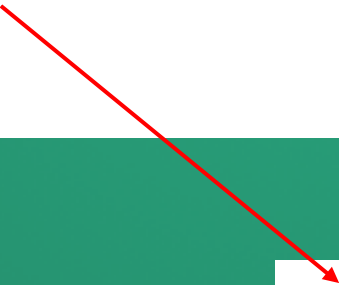
```
self.total = ko.computed(function() {  
    return parseInt(self.val1()) +  
        parseInt(self.val2());  
}, self);
```



Dependency Tracking

subscriptions

```
self.val.subscribe(function(newVal) {  
    self.save_value(self);  
}).bind(self);
```

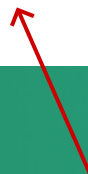


```
apex.server.process(...)
```



Templating

```
// Static template name  
<div data-bind="template: { name: 'personListTemplate', data: employees }"></div>
```



Dynamic Template

```
// Template determined by a function  
<div data-bind="template: { name: selectedTemplate} "></div>
```





Templating

```
<script type="text/ko-template" id="personListTemplate">
<ul data-bind="foreach: people">
  <li data-bind="text: name"></li>
</ul>
</script>
```

```
<script type="text/ko-template" id="personTableTemplate">
<table>
<thead><tr><th>Name</th><th>Salary</th></tr></thead>
<tbody data-bind="foreach: people">
<tr>
  <td><span data-bind="text: name"></span></td>
  <td><span data-bind="text: sal"></span></td>
</tr>
</tbody>
</table>
</script>
```

KnockoutJS and APEX

APEX 5.1

```
// APEX 5.1  
#IMAGE_PREFIX#libraries/oraclejet/2.0.2/js/libs/knockout/knockout-3.4.0.js
```

knockoutjs.com/downloads/index.html

Include the library

Page 10: Simple Data Bind

Pre-Rendering

Regions

Content Body

Work Week

Attributes

Items

P10_VAL1

P10_VAL2

P10_TOTAL

Post-Rendering

Filter Properties

JavaScript

File URLs

```
#IMAGE_PREFIX#libraries/oraclejet/2.0.2/js/libs/knockout/knockout-3.4.0.js
```

Function and Global Variable Declaration

```
function workWeek() {  
  var self = this;  
  self.val1 = ko.observable(0);  
  self.val2 = ko.observable(0);  
  
  self.total = ko.computed(function() {  
    return parseInt(self.val1()) +  
      parseInt(self.val2());  
  }, self);  
}
```

Execute when Page Loads

```
var viewModel = new workWeek();  
ko.applyBindings(viewModel);
```

Attributes

▼ Items

I P10_VAL1

I P10_VAL2

A≡ P10_TOTAL

▼ Advanced

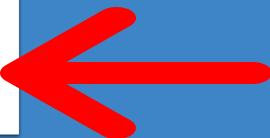
CSS Classes

Custom Attributes

data-bind="textInput: val1"

^

^



Attributes

▼ Items

I P10_VAL1

I P10_VAL2

A≡ P10_TOTAL

▼ Advanced

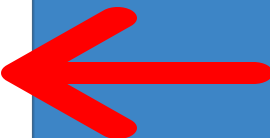
CSS Classes

Custom Attributes

data-bind="text: total"

^

^



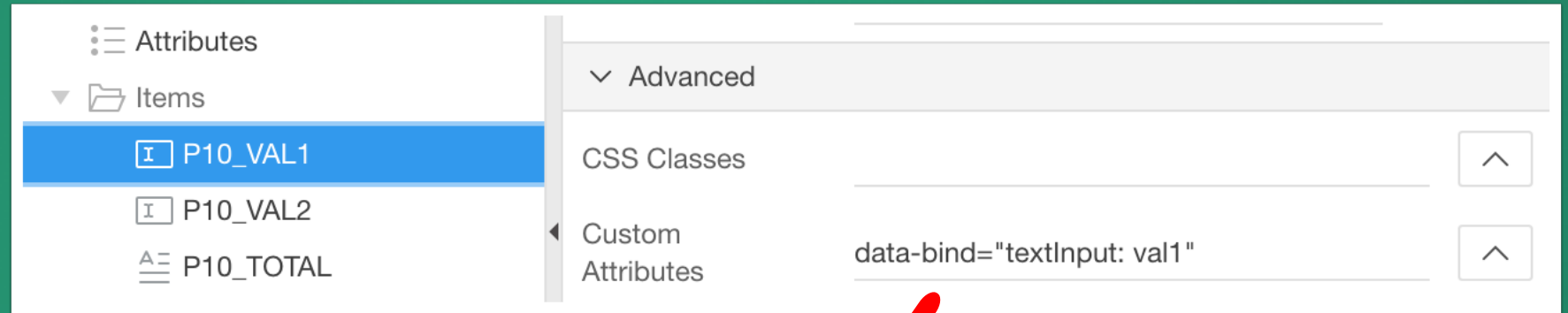
ko pre-rendered plugin

Playing Nice with APEX

github.com/ErikSchierboom/knockout-pre-rendered

```
#APP_IMAGES#knockout-pre-rendered.min.js
```

Add "init" binding



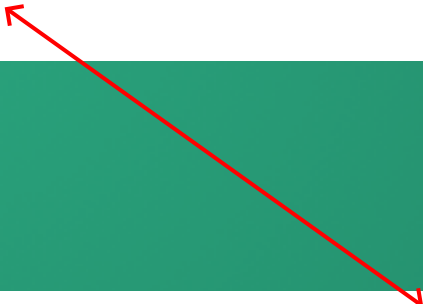
```
data-bind="init, textInput: val1"
```

Add "foreachInit" binding for looping

```
data-bind="foreachInit: products"
```

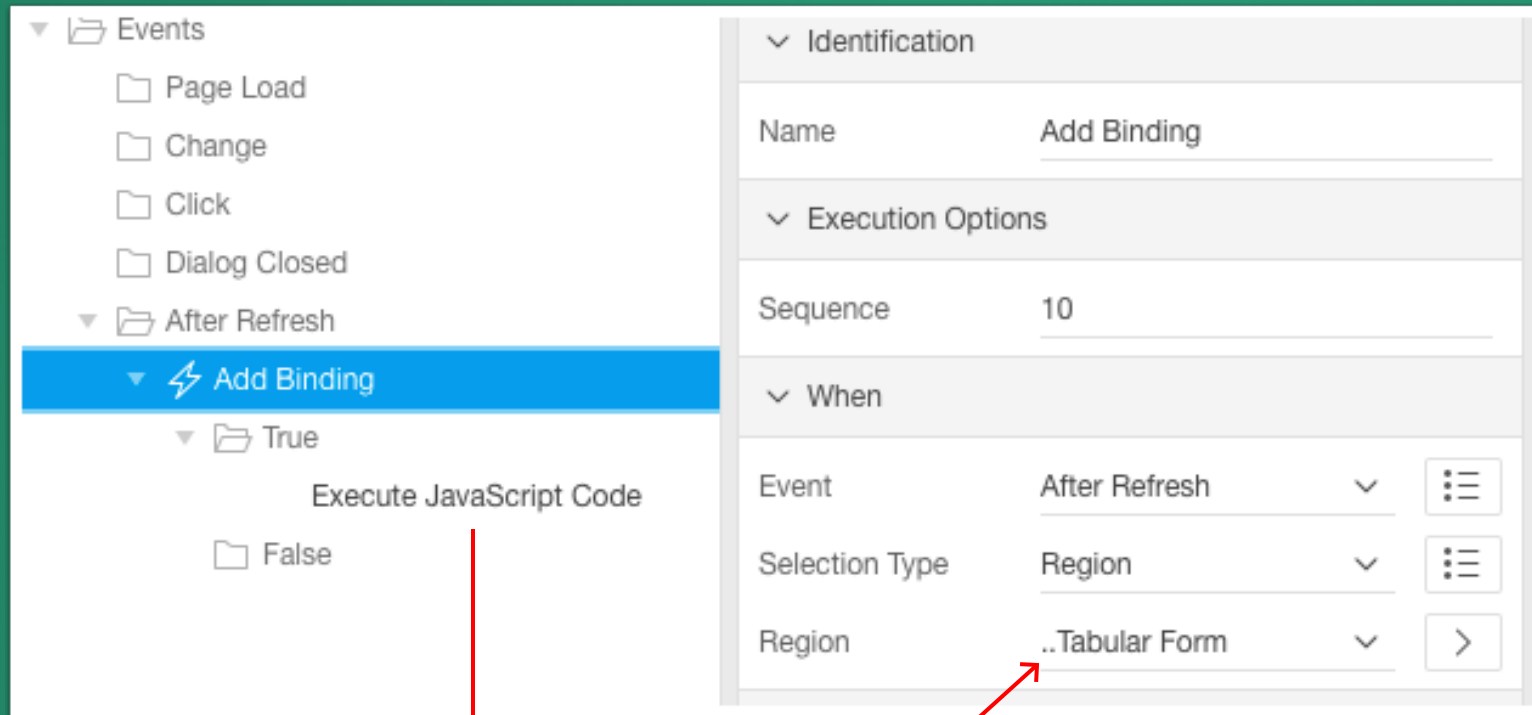
```
// or with template somewhere else on page
```

```
data-bind="foreachInit: { name: 'productTemplate', data: products }"
```



```
<script type="text/ko-template" id="productTemplate">  
  <li data-bind="text: name"></li>  
</script>
```

Adding "foreachInit" to the correct parent



```
var $rowContainer = $(this.triggeringElement).find(".t-Report-report tbody");  
  
// Add ko foreach bind  
$rowContainer.attr("data-bind",  
    "foreachInit: { name: 'productTemplate', data: products }");
```

Demos

Q&A

rimblas.com/blog/presentations
github.com/rimblas

Deliver a Knockout with your APEX Applications

Jorge Rimblas 

Twitter: @rimblas