

Demystifying the APEX JavaScript API-s

@stefan__dobre

#techfest19

Demo: <https://github.com/stefandobre/apex-js-api-talk>

FOEX Plug-in Framework

- 160+ Plug-ins
- Single Page Applications
- Drag & Drop
- Web Sockets

FOEX Plug-in Framework

- 160+ Plug-ins
- Single File
- Drag & Drop
- Web Socket

FOEX Window [Plug-In]	▲
D3 Collapsible Tree Chart [Plug-In]	
D3 Treemap Chart [Plug-In]	
Data Upload Column Mapping	
FOEX Application Settings [Plug-In]	
FOEX Big Number [Plug-In]	
FOEX Calendar [Plug-In]	
FOEX Chart [Plug-In]	
FOEX Chart Series [Plug-In]	
FOEX CKEditor Viewer [Plug-In]	
FOEX Code Editor [Plug-In]	
FOEX Content Loader Region [Plug-In]	
FOEX Desktop [Plug-In]	
FOEX Documentation Viewer [Plug-In]	
FOEX Draw Component [Plug-In]	
FOEX Dropzone File Upload [Plug-In]	
FOEX Field Container [Plug-In]	
FOEX Fieldset [Plug-In]	
FOEX Flex Paper [Plug-In]	
FOEX Form [Plug-In]	
FOEX Form Dynamic [Plug-In]	
FOEX Google Map [Plug-In]	
FOEX Grid [Plug-In]	
FOEX Layout [Plug-In]	
FOEX List View (Deprecated) [Plug-In]	
FOEX List View (Editable) [Plug-In]	
FOEX Navigation [Plug-In]	
FOEX Page Settings [Plug-In]	
✓ FOEX Pane [Plug-In]	
FOEX PLSQL Region [Plug-In]	
FOEX Region Spacer [Plug-In]	
FOEX Splitter (for HBOX/VBOX Layouts) [Plug-In]	
FOEX Static Region [Plug-In]	
FOEX Toolbar [Plug-In]	
FOEX Tree [Plug-In]	
FOEX Tree Grid [Plug-In]	
FOEX Twitter [Plug-In]	
FOEX Viewport [Plug-In]	
FOEX Websocket Settings [Plug-In]	
FOEX Wizard Progress [Plug-In]	
-	▼

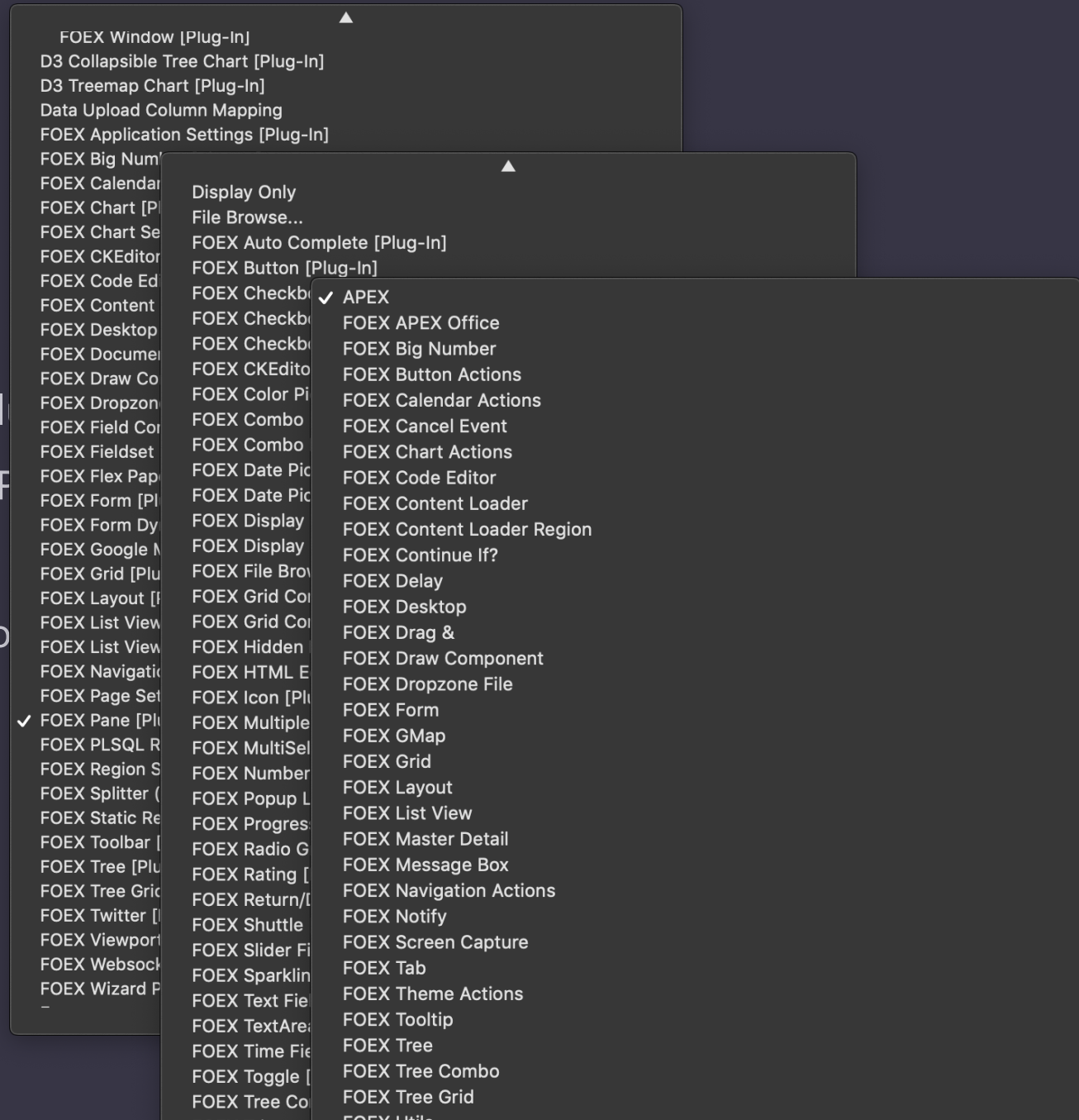
FOEX Plug-in Framework

- 160+ Plug-ins
- Single File
- Drag & Drop
- Web Software



FOEX Plug-in Framework

- 160+ Plug-ins
- Single File
- Drag & Drop
- Web Software

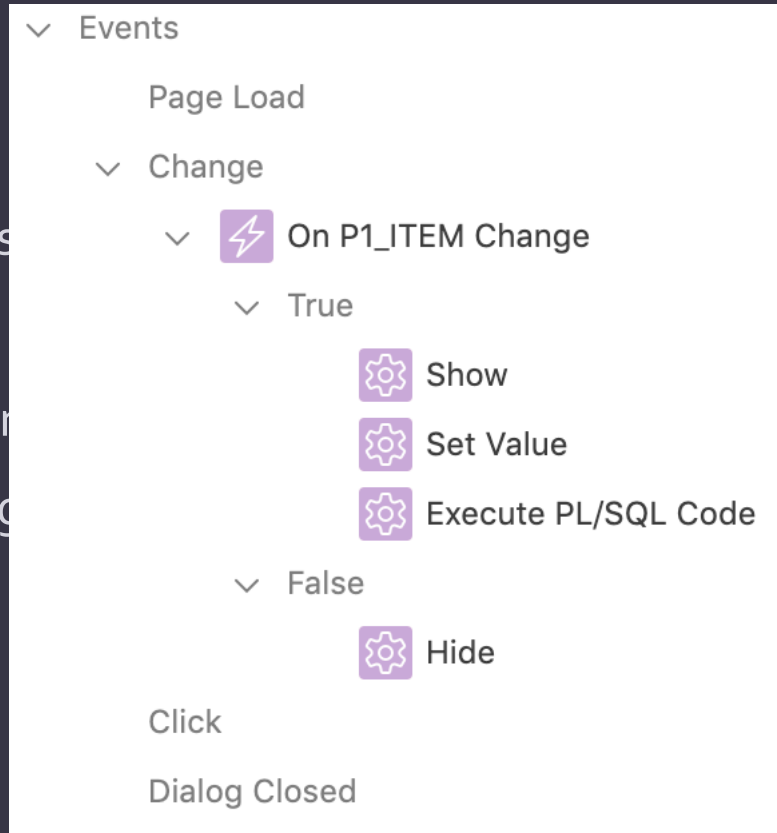


A case for JavaScript in APEX

- Build more dynamic apps
- Improve user experience
- For when Dynamic Actions are not enough
- Leverage already existing functionality
- JavaScript will only keep growing

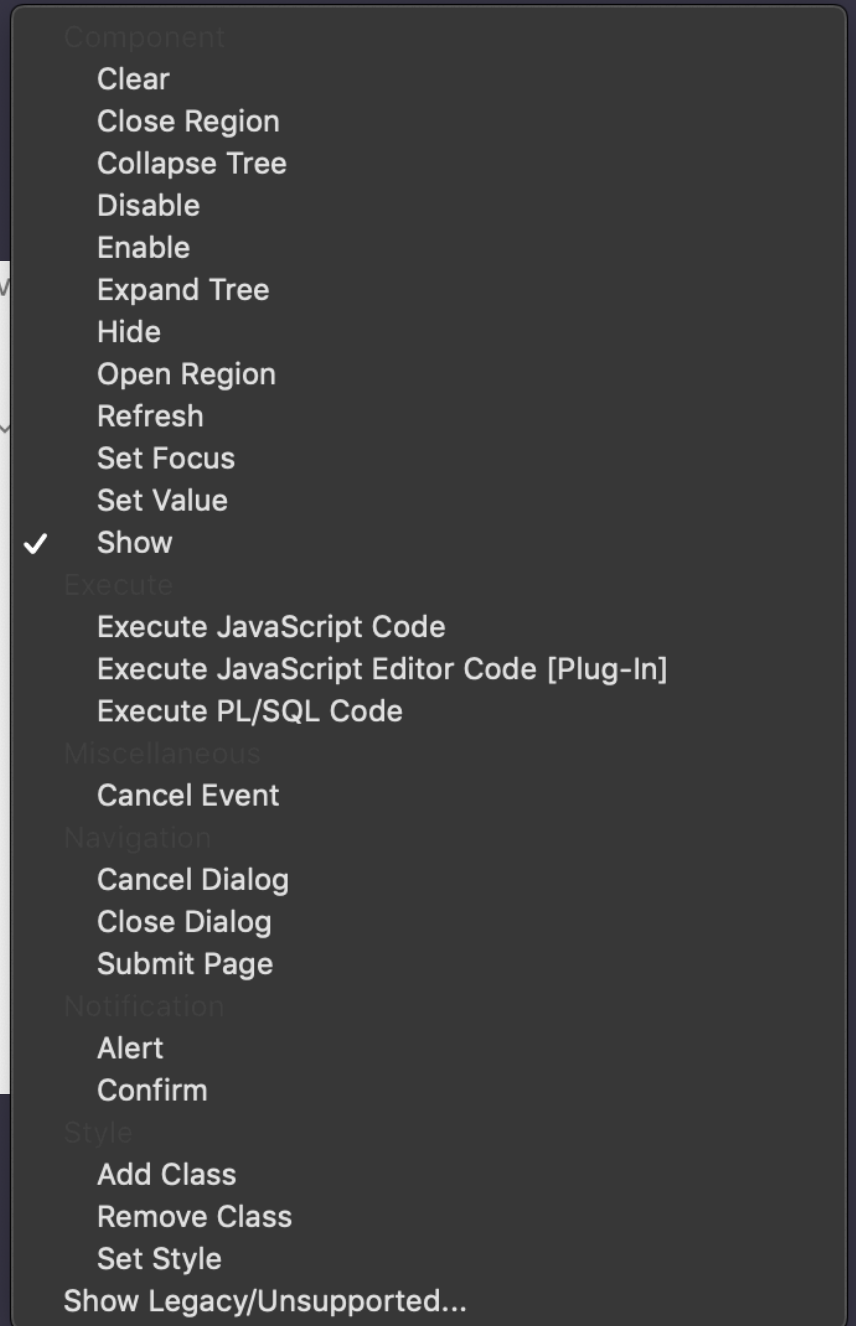
A case for JavaScript in APEX

- Build more dynamic apps
- Improve user experience
- For when Dynamic Action
- Leverage already existing
- JavaScript will only keep



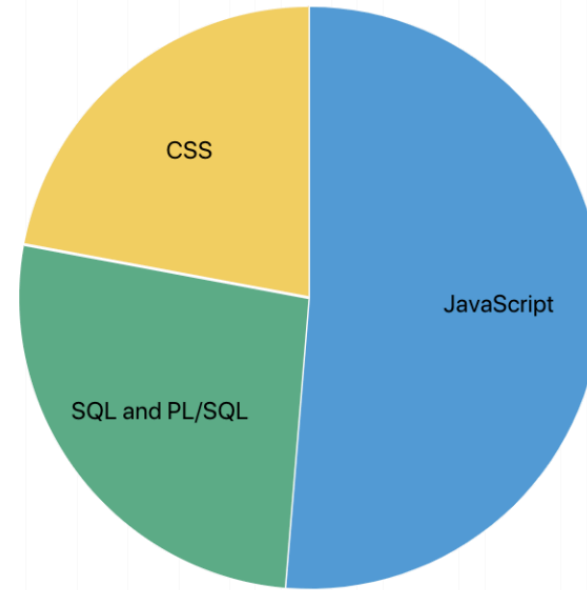
A case for JavaScript in APEX


- Build more dynamic apps
- Improve user experience
- For when Dynamic Action
- Leverage already existing
- JavaScript will only keep



A case for JavaScript in APEX

- Build
- Impro
- For w
- Lever
- JavaS



Language 	Code Lines
CSS	378,818
JavaScript	880,774
SQL and PL/SQL	457,198
	1,716,790

Component
Clear
Close Region
Collapse Tree

Add Class
Remove Class
Set Style
Show Legacy/Unsupported...

Word of caution



Writing custom code =

- Longer development
- Longer maintenance
- Statistically, more bugs

What JavaScript API-s does APEX offer us?

- JavaScript API Reference (apex.oracle.com/jsapi)
- Universal Theme JavaScript (apex.oracle.com/ut)
- Integration with the API-s of 3rd party components
 - E.g. Charts, Calendar, Rich Text Editor

APEX JavaScript API Reference

apex.oracle.com/jsapi

1.
Widgets

2.
Namespaces

3.
Interfaces

4.
Non-Namespace
API-s

APEX JavaScript API Reference

apex.oracle.com/jsapi

1.

Widgets

2.

Namespaces

3.

Interfaces

2. Namespaces

```
apex
apex.actions
apex.da
apex.debug
apex.event
apex.item
apex.lang
apex.message
apex.model
apex.navigation
apex.navigation.dialog
apex.navigation.popup
apex.page
apex.region
apex.server
apex.storage
apex.theme
```

- *Could* be compared to PL/SQL packages
- Utility functions grouped together by purpose and scope
- **Note:** `apex.item` & `apex.region` are functions themselves!

demo

3. Interfaces

```
actions  
htmlBuilder  
interactiveGridView  
item  
model  
region  
treeNodeAdapter
```

- We cannot call interface methods directly
- Components implement interfaces
- Access the methods through their respective component
 - e.g an item interface's methods can be invoked through `apex.item('P1_ITEM')`

3. Interfaces

```
actions  
htmlBuilder  
interactiveGridView  
item  
model  
region  
treeNodeAdapter
```

- `apex.region('emp').call('getActions')`
- `apex.util.htmlBuilder()`
- `apex.region('emp').call('getCurrentView')`
- `apex.item('P1_ITEM')`
- `apex.region('emp').call('getCurrentView').model`
- `apex.region('emp')`
- `apex.region('emp').call('getNodeAdapter')`

demo

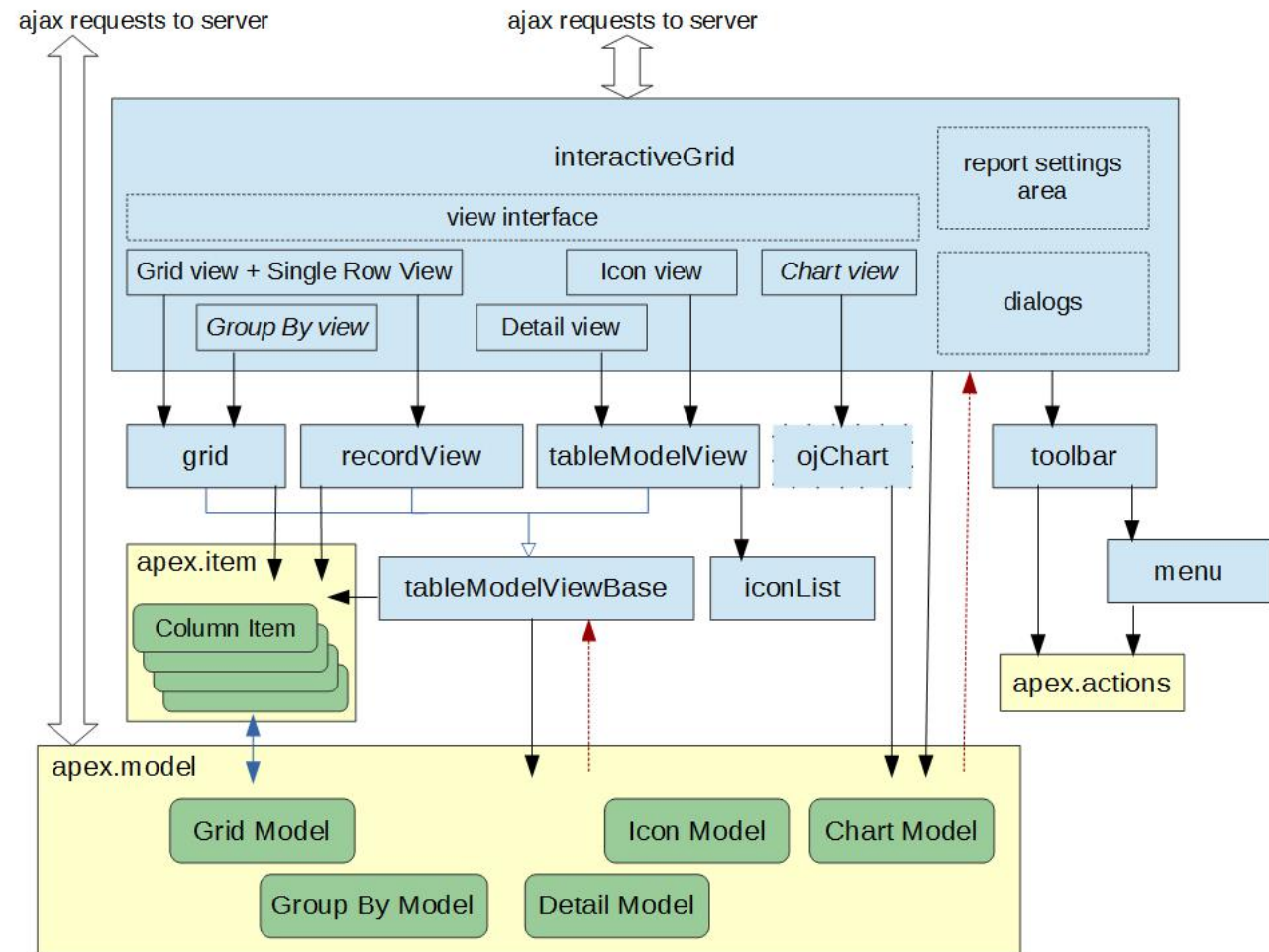
1. Widgets

```
grid
iconList
interactiveGrid
menu
menuButton
recordView
tableModelView
tableModelViewBase
treeView
```

- Help us control 3 APEX components:
 - Tree
 - Menu
 - Interactive Grid
- Tree & IG
 - Implement the region interface
 - `apex.region('emp')`
 - The interface controls a jQuery UI widget
 - `apex.region('emp').call('functionName')`
 - `apex.region('emp').call('option', 'optionName')`
 - `apex.region('emp').call('option', 'optionName', 'optionValue')`
- Menu
 - Regular jQuery UI widget
 - `$('#menuId').menu('functionName')`
 - `$('#menuId').menu('option', 'optionName')`
 - `$('#menuId').menu('option', 'optionName', 'optionValue')`

demo

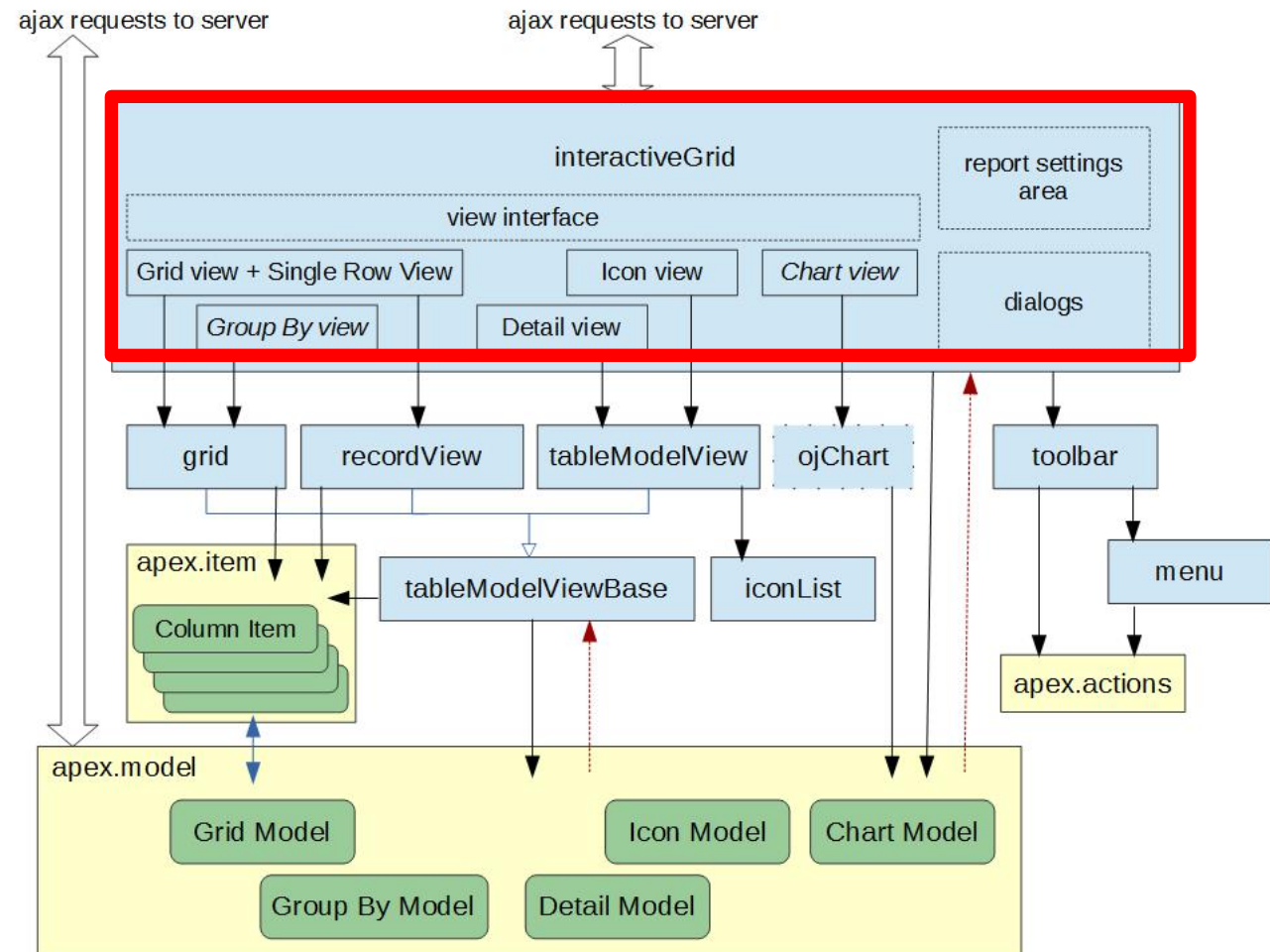
grid Widget



<https://hardlikesoftware.com/weblog/2016/06/08/interactive-grid-under-the-hood/>

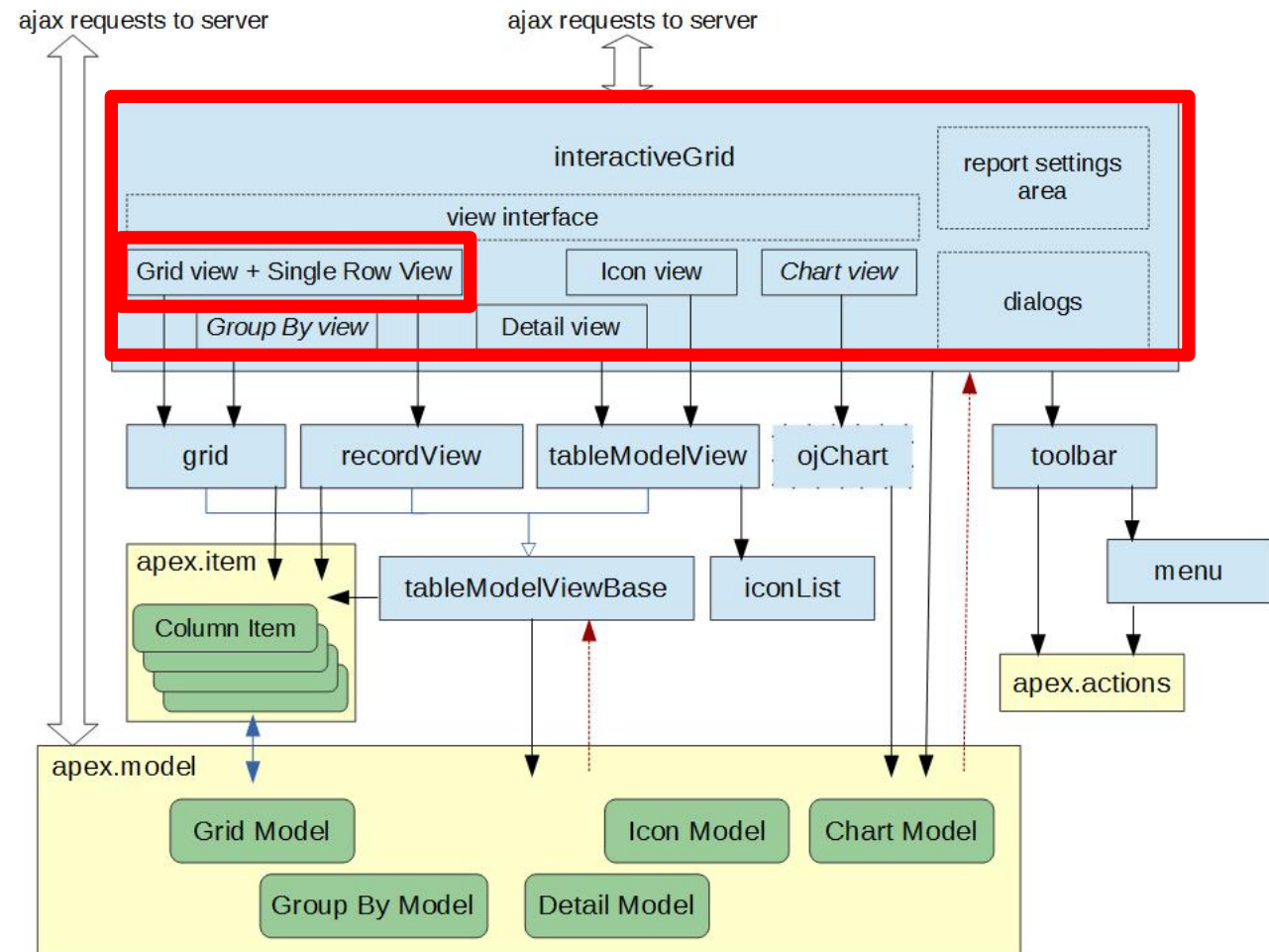
grid Widget

```
var region = apex.region('emp');
```



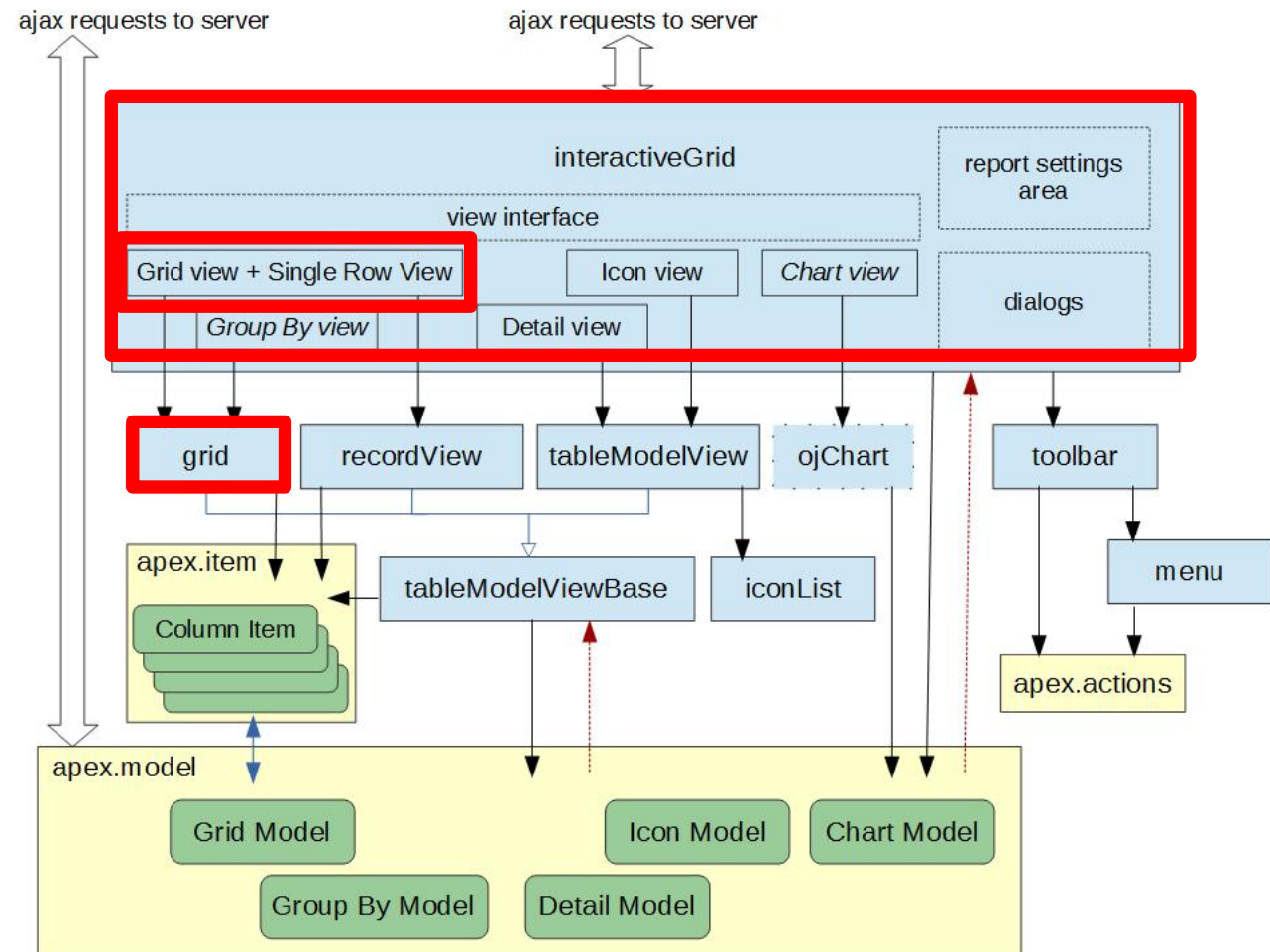
grid Widget

```
var region      = apex.region('emp');
var gridView   = region.call('getViews', 'grid');
```



grid Widget

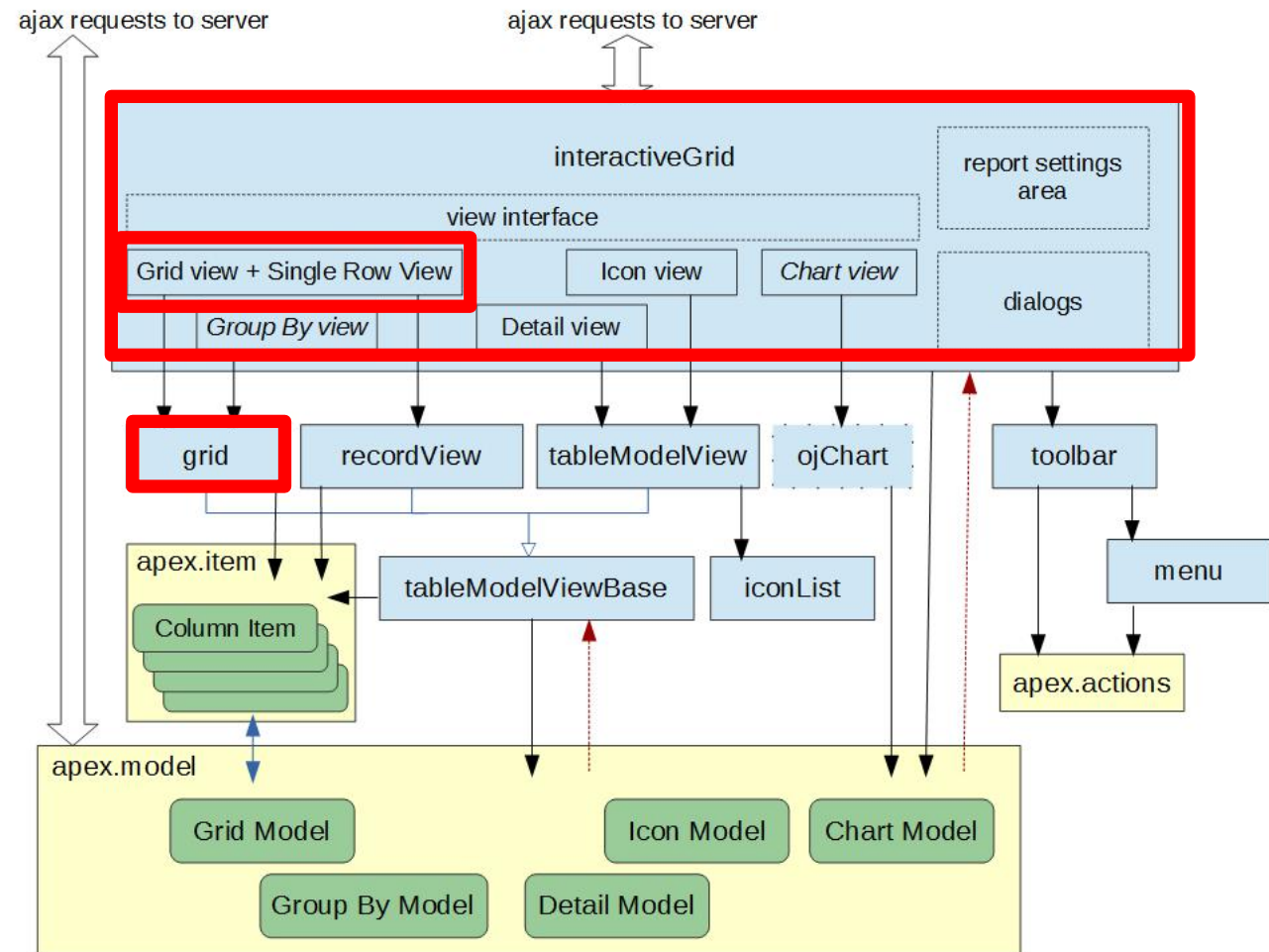
```
var region      = apex.region('emp');  
var gridView   = region.call('getViews', 'grid');  
var gridWidget = gridView.view$;
```



<https://hardlikesoftware.com/weblog/2016/06/08/interactive-grid-under-the-hood/>

grid Widget

```
var region      = apex.region('emp');  
var gridView   = region.call('getViews', 'grid');  
var gridWidget = gridView.view$;  
  
gridWidget.grid('selectAll');
```



<https://hardlikesoftware.com/weblog/2016/06/08/interactive-grid-under-the-hood/>

Some thoughts

- Very useful, but there is a learning curve
- Use with caution
- Explore for yourself