

PRETIUS

Drilling into the Iceberg of APEX Interactive Grid

Adam Kierzkowski

 a_kierzkowski

HrOUG 2021

ABOUT ME



Oracle APEX Developer

Working in international teams, experienced in building global applications with use of **PL/SQL, JavaScript and APEX**

APEX Plug-ins creator

I develop new and maintain the existing **Pretius APEX Plug-ins** shared to the community at apex.world

Conference speaker

Sharing my APEX knowledge and passion at different conferences and meet-ups around the world

Amateur photographer

Taking some pictures in a free time

INTERACTIVE GRID – WHAT’S WORNG WITH YOU ?

Q Search: All Text Columns Go Actions Edit Save Add Row

Salary greater than 1000
Highlighted

	Employee Id	First Name	Last Name	Email	Phone Number	Hire Date	Job Id
<input type="checkbox"/>	100	Steven	King	SKING	515.123.4567	6/17/2003	AD_PRES
<input checked="" type="checkbox"/>	101	Neena	Kochhar	NKOCHHAR	515.123.4568	9/21/2005	AD_VP
<input type="checkbox"/>	102	Lex	De Haan	LDEHAAN	515.123.4569	1/13/2001	AD_VP
<input type="checkbox"/>	103	Alexander	Hunold	AHUNOLD	590.423.4567	1/3/2006	IT_PROG
<input type="checkbox"/>	104	Bruce	Ernst	BERNST	590.423.4568	5/21/2007	IT_PROG
<input type="checkbox"/>	105	David	Austin	DAUSTIN	590.423.4569	6/25/2005	IT_PROG
<input type="checkbox"/>	106	Valli	Pataballa	VPATABAL	590.423.4560	2/5/2006	IT_PROG
<input type="checkbox"/>	107	Diana	Lorentz	DLORENTZ	590.423.5567	2/7/2007	IT_PROG
<input type="checkbox"/>	108	Nancy	Greenberg	NGREENBE	515.124.4569	8/17/2002	FI_MGR
<input type="checkbox"/>	109	Daniel	Faviet	DFAVIET	515.124.4169	8/16/2002	FI_ACCOUNT
<input type="checkbox"/>	110	John	Chen	JCHEN	515.124.4269	9/28/2005	FI_ACCOUNT
<input type="checkbox"/>	111	Ismael	Sciarra	ISCIARRA	515.124.4369	9/30/2005	FI_ACCOUNT
<input type="checkbox"/>	112	Jose Manuel	Urman	JMURMAN	515.124.4469	3/7/2006	FI_ACCOUNT
<input type="checkbox"/>	113	Luis	Popp	LPOPP	515.124.4567	12/7/2007	FI_ACCOUNT



OLD BUT GOOD!

Report Columns

	Displayed	Label	Column
	✓	Id	
	✓	Rank	
	✓	Urban Area	
	✓	Country	
	✓	Population	
	✓	Area	
	✓	Density	
	✓	Sources	
	✓	Continent	

Edit Column

Column **RANK**

Type **NUMBER**

* Label

Format Mask

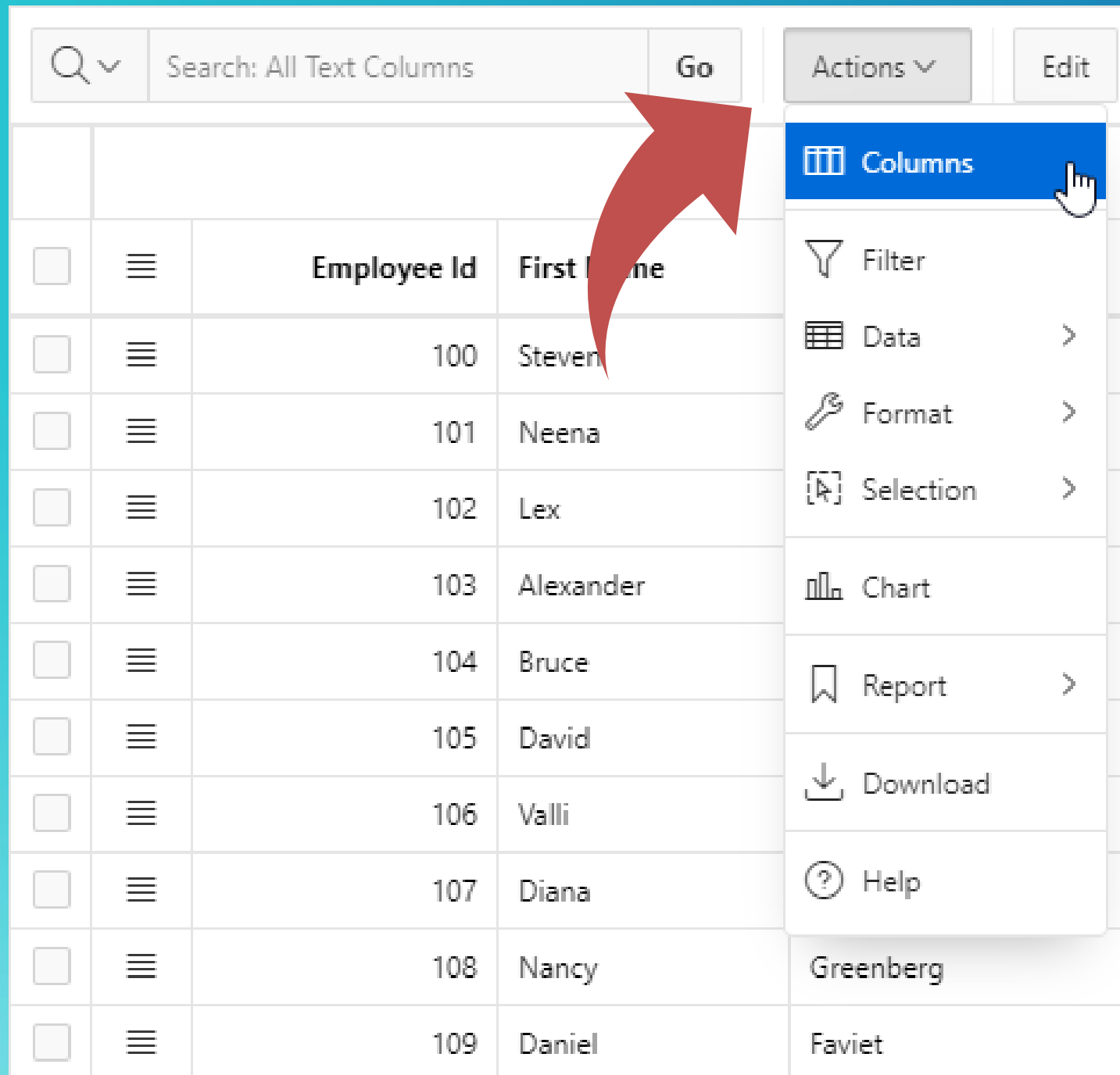
Allow Users To

- Hide
- Sort
- Filter
- Highlight
- Control Break
- Aggregate
- Compute
- Chart
- Group By

Target

Cancel

BLINK IF IT HAPPENED TO YOU 😊



The screenshot shows a data grid with columns for 'Employee Id' and 'First Name'. The 'Actions' menu is open, showing options like 'Columns', 'Filter', 'Data', 'Format', 'Selection', 'Chart', 'Report', 'Download', and 'Help'. A red arrow points from the 'Columns' option in the menu to the 'Go' button in the search bar.

		Employee Id	First Name
<input type="checkbox"/>	☰	100	Steven
<input type="checkbox"/>	☰	101	Neena
<input type="checkbox"/>	☰	102	Lex
<input type="checkbox"/>	☰	103	Alexander
<input type="checkbox"/>	☰	104	Bruce
<input type="checkbox"/>	☰	105	David
<input type="checkbox"/>	☰	106	Valli
<input type="checkbox"/>	☰	107	Diana
<input type="checkbox"/>	☰	108	Nancy
<input type="checkbox"/>	☰	109	Daniel

*customer : I need a simple editable report

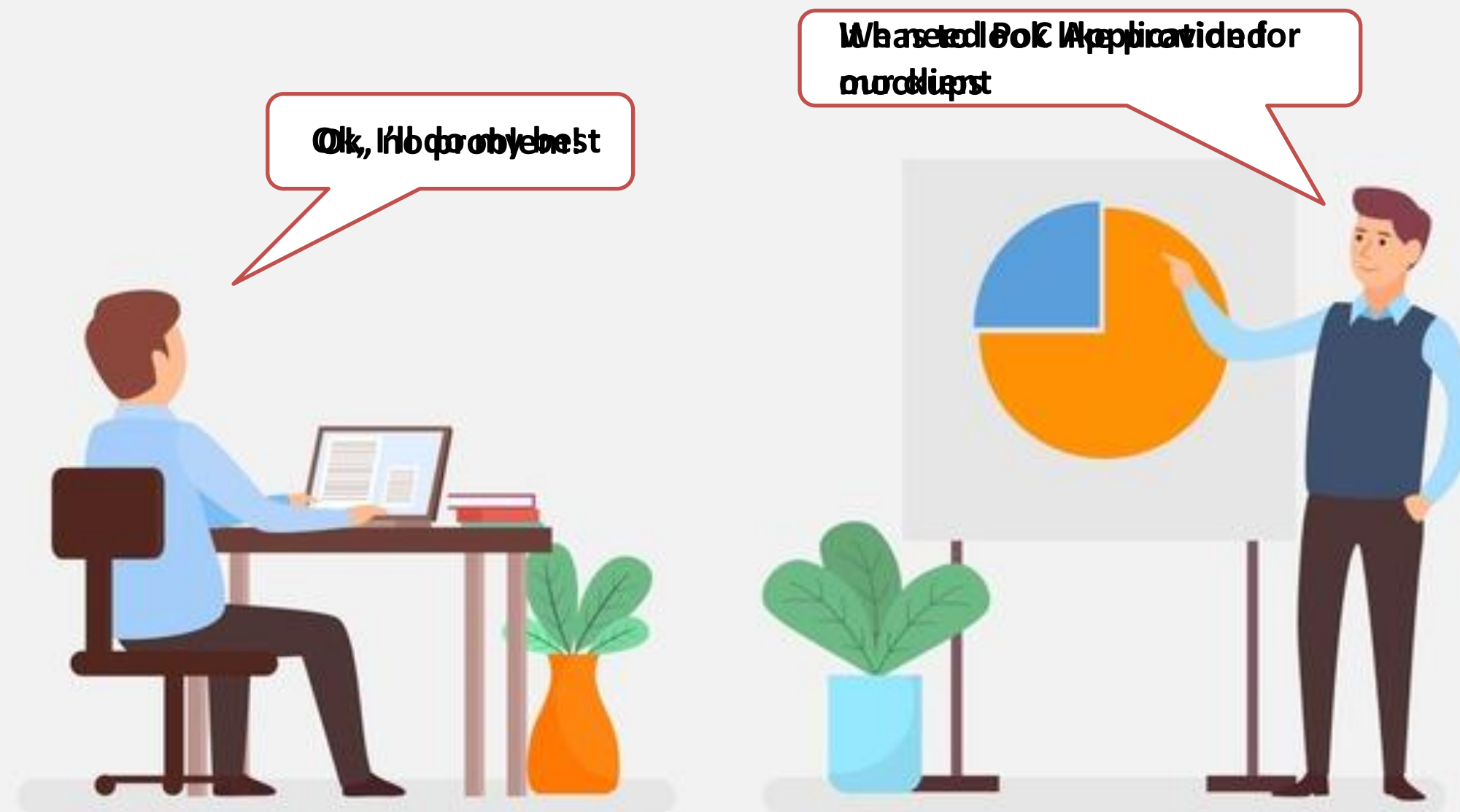
*me : <building an interactive grid>

*customer : What is that? Remove it

A FEW MOMENTS LATER...

***customer: I have clicked something wrong, and a column disappeared.**

SO WHAT HAS CHANGED?



SO WHAT HAS CHANGED?

YouTube PL Szukaj

APEX connect

WHY ABOUT JET CHARTS?

Ok, no problem!

We need graphs for our client

pretius

Adam Kierzkowski: "Fancy Charts and Dashboards: Customizing Oracle Jet in APEX"

352 wyświetlenia • 11 lut 2021

11 0 UDOŚTĘPNIJ ZAPISZ

doagtv 1,3 tys. subskrypcji

Adam Kierzkowski zeigt bei der APEX connect 2020 [Online], welche Möglichkeiten das APEX-Tool Jet Charts bietet.

SUBSKRYBUJ



akierzkowski



a_kierzkowski



Fancy charts and dashboards Customizing JET Charts in APEX

Drilling into the iceberg of APEX Interactive Grid

“

Working in IT is not
stressing at all

Adam, Age 26

ABOUT PRETIUS

A screenshot of a presentation slide showing the Pretius logo. The logo consists of the word "pretius" in a lowercase, sans-serif font. The "pre" is in black, and the "tius" is in a light blue color. The background of the slide is a light grey gradient.

pretius

WHO ARE WE

Team of over 200 IT experts, including over **35 Low-code developers**

WHAT WE OFFER

Outsourcing remote teams and software development process.

WHERE WE ARE

We are acting **globally** from our office in Warsaw, Poland.

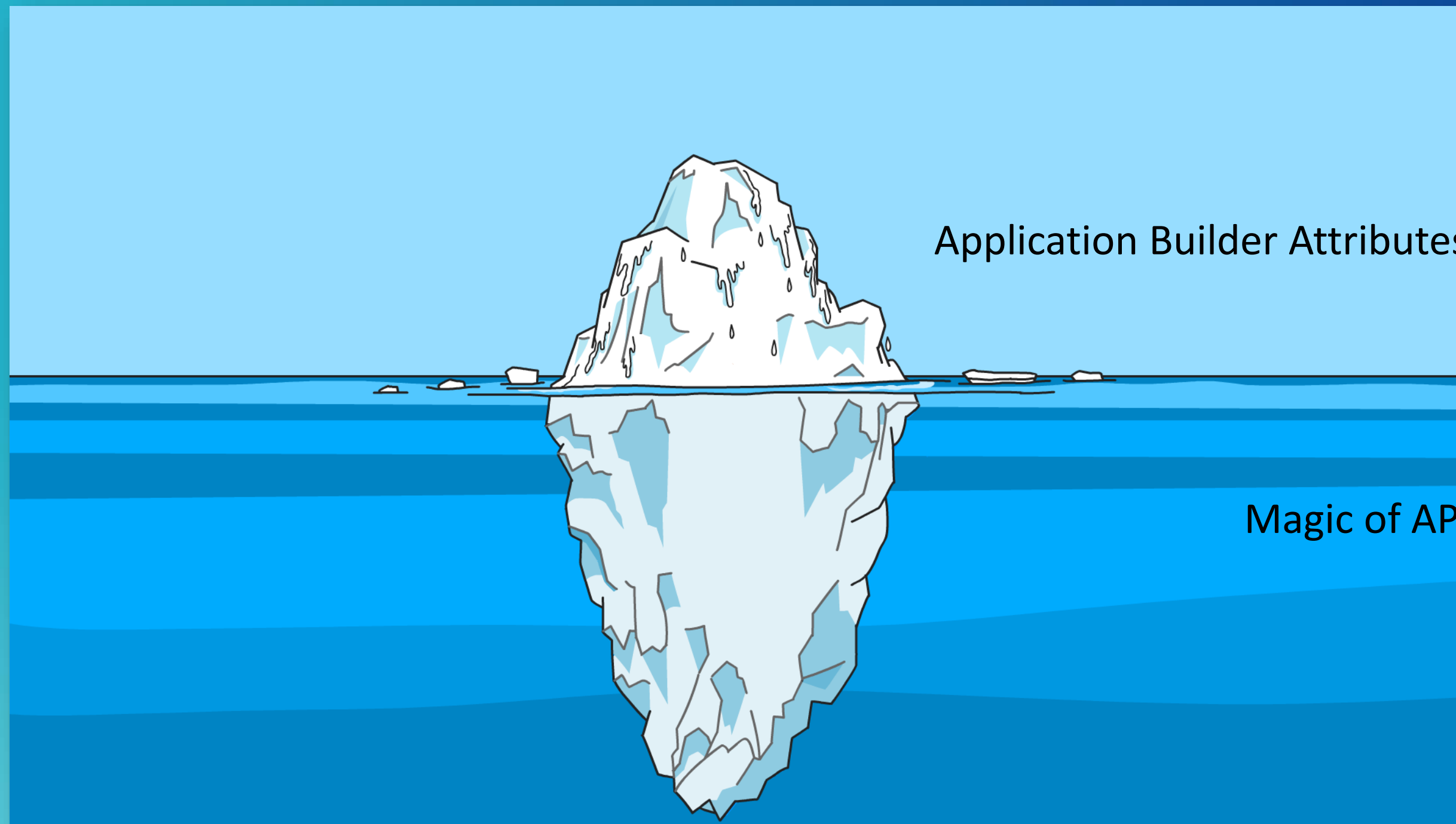
APEX EXPERTS

Check our **APEX plug-ins** and **translate-apex.com** project.

LET'S GET BACK TO MY CHALLENGE

					2020												2021			
Workpackage ↑±2	Department	Role	Local Cost Center	Site	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul
▼ Theme/Project: 78987 - Flamingo																				
WPCK1				MILESTONE	★															
	▶			<input checked="" type="checkbox"/> PREDICTIVE DEMANDS	1.7	1.5	2.1	2.1	2.2	3.5	3.8	3.8	3.8	4.2	4.2	4.2	4.2	4.2	4.2	4.2
				TOTAL WF ALLOCATION	2.8	2.4	3.4	3.4	3.5	5.8	6.2	6.2	6.2	6.8	6.7	6.7	6.7	6.7	6.6	4.2
HRD	CL		12383	Netherlands, Amsterdam																
HRD	MGR		12383	Netherlands, Amsterdam																
ODS2	A		TEST	TEST	0.6	0.6	0.8	0.8	0.8	1.2	1.4	1.4	1.4	1.6	1.7	1.7	1.7	1.7	1.8	1.8
MDSL1	DEV		12374	United Kingdom, Liverpool																
MD	MGR		12371	USA, Atlanta																
ODS2	CL		12364	USA, New York	1.1	0.9	1.3	1.3	1.3	2.3	2.4	2.4	2.4	2.6	2.5	2.5	2.5	2.5	2.4	
ODS2	CL		TEST	TEST	1.1	0.9	1.3	1.3	1.3	2.3	2.4	2.4	2.4	2.6	2.5	2.5	2.5	2.5	2.4	2.4
WPCK3				MILESTONE																
	▶			<input checked="" type="checkbox"/> PREDICTIVE DEMANDS	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8
				TOTAL WF ALLOCATION	1.1	1.1	1.1	0.7	0.7	0.7	0.7	0.7	0.7	0.1	0.1	0.1	0.0	0.0	0.0	0.0
HRD	A		12357	Greece, Athen				0.7	0.7	0.7	0.7	0.7	0.1	0.1	0.1					
MDS1	A		12359	France, Lyon	1.1	1.1	1.1													

WHAT'S THE PLAN?



TOUGH BEGINNINGS

The image displays three overlapping screenshots from a development environment:

- Top Screenshot (DevTools Console):** Shows the 'Console' tab with a log entry: `> apex.region('EMPLOYEE_FTE').call('getViews')`.
- Bottom-Left Screenshot (Code Editor):** Shows a JavaScript function definition for `options`. It includes a `myContextMenu` object with an `items` array containing two action items. The first item has `type: "action", icon: "fa-clone", label: "Duplicate", action: "selection-d"`. The second item has `type: "action", icon: "fa-calendar-u"`.
- Bottom-Middle Screenshot (Code Editor):** Shows a code editor with two conditional blocks. The first block checks `$v('P200_GRANULARITY') == 'q'` and calls `apex.region('COST_CENTERS').call('get')`. The second block checks `$v('P200_GRANULARITY') == 'm'` and calls `apex.region('COST_CENTERS_MONTH').cal`.
- Right Screenshot (Configuration Panel):** Shows a configuration panel with tabs for 'Region' and 'Attributes'. A search bar contains the text 'Advanced'. Below the search bar, there is a section for 'Advanced' with a dropdown menu for 'Oracle Text Index Column' set to '- Select -', a text input for 'Email From Address', and a 'JavaScript Initialization Code' section containing `customGrid.customGridConfig`.

CHEAT SHEETS FOR HELP

Get Widget

Before APEX 18.1 you should use:

```
apex.region("emp").widget
```

instead of

```
apex.region("emp").call
```

Get Record

```
var grid = apex.region  
var model = grid.model;  
var record = model.getRe
```

Get Grid Options

```
apex.region("emp").call("
```

Get Model Options

```
apex.region("emp").call("
```

Initial Grid Configuration

```
function(config) {  
  
    if (!config.toolbar) {  
        config.toolbar = {};  
    }  
  
    config.toolbar.searchField = false; // hide t  
  
    config.toolbar.actionMenu = false; // hide to  
  
    // disable column reorder  
    config.views.grid.features.reorderColumns = f  
  
    // enable selection persistence across differ  
    config.views.grid.features.persistSelection =  
  
    // hide grid footer  
    config.views.grid.features.footer = false;  
  
    // disable initial selection  
    config.initialSelection = false;
```



Marko Gorički

 mgoricki



Stefan Dobre

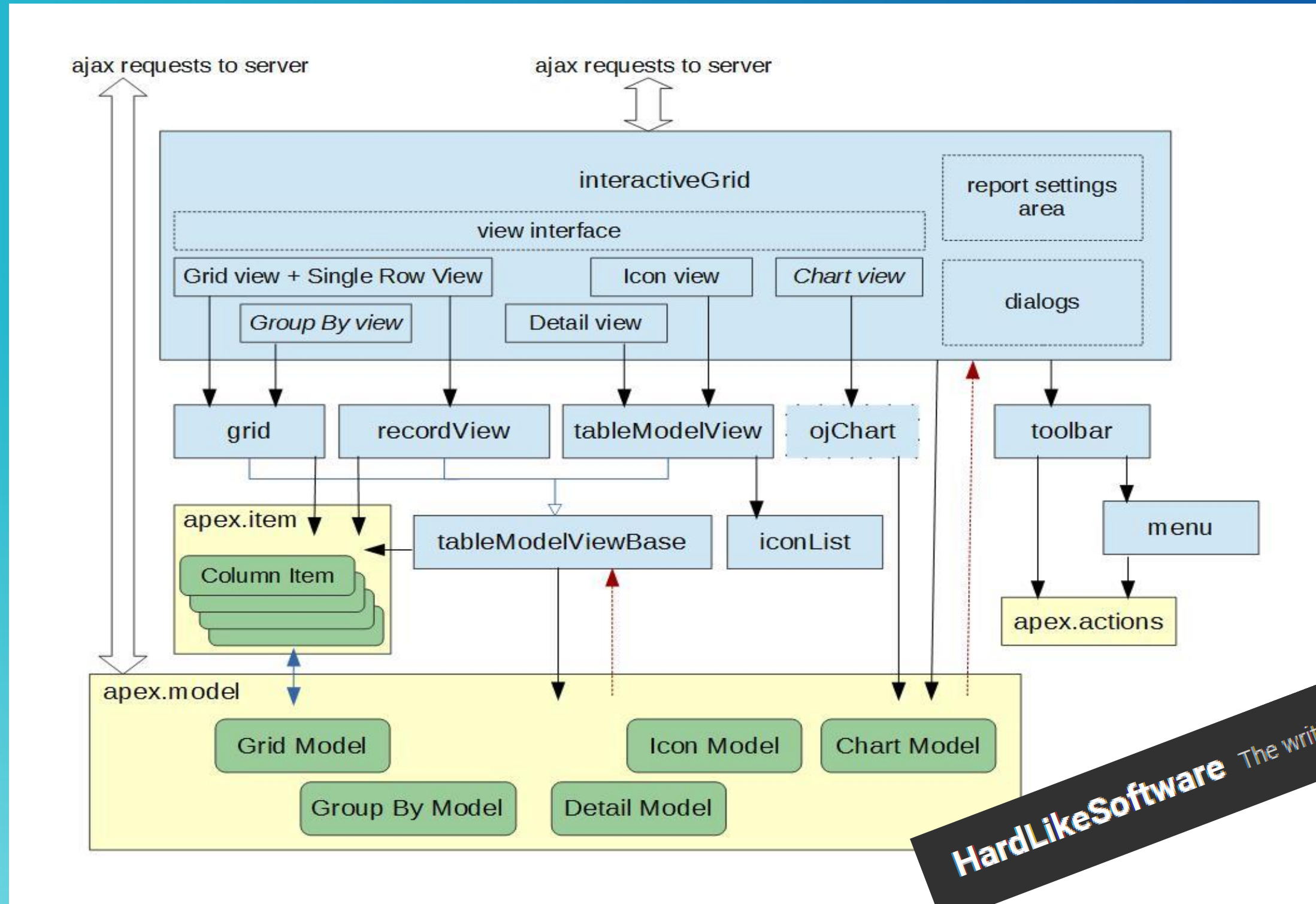
 stefandobre

IS THIS ICEBERG MADE OF ICE?

The screenshot shows a search interface with a search bar containing the text "Search: All Text Columns" and a "Go" button. To the right of the search bar are three view toggle icons: a grid icon (selected), a table icon, and a list icon. Below the search bar, there are five search results, each consisting of a person icon and a name: ADAMS, KING, BLAKE, JAMES, and MILLER.

Icon	Name
	ADAMS
	KING
	BLAKE
	JAMES
	MILLER

PICTURE THAT



HardLikeSoftware The writings of John Snyders, mostly about software.

INTERACTIVE GRID WIDGET

interactiveGrid

WIDGET NAME

```
> interactiveGrid
```

WIDGET INITIALIZATION

```
> $('div#STATIC_ID').interactiveGrid();
```

WIDGET INITIALIZATION WITH OPTIONS

```
> $('div#STATIC_ID').interactiveGrid({  
  "option1": val_1,  
  "option2": val_2  
});
```

CALLING WIDGET'S METHOD

```
> $('div#STATIC_ID').interactiveGrid("getViews");
```

WHERE'S OUR WIDGET INSTANCE?



```
$('#div#STATIC_ID').interactiveGrid("getViews");
```

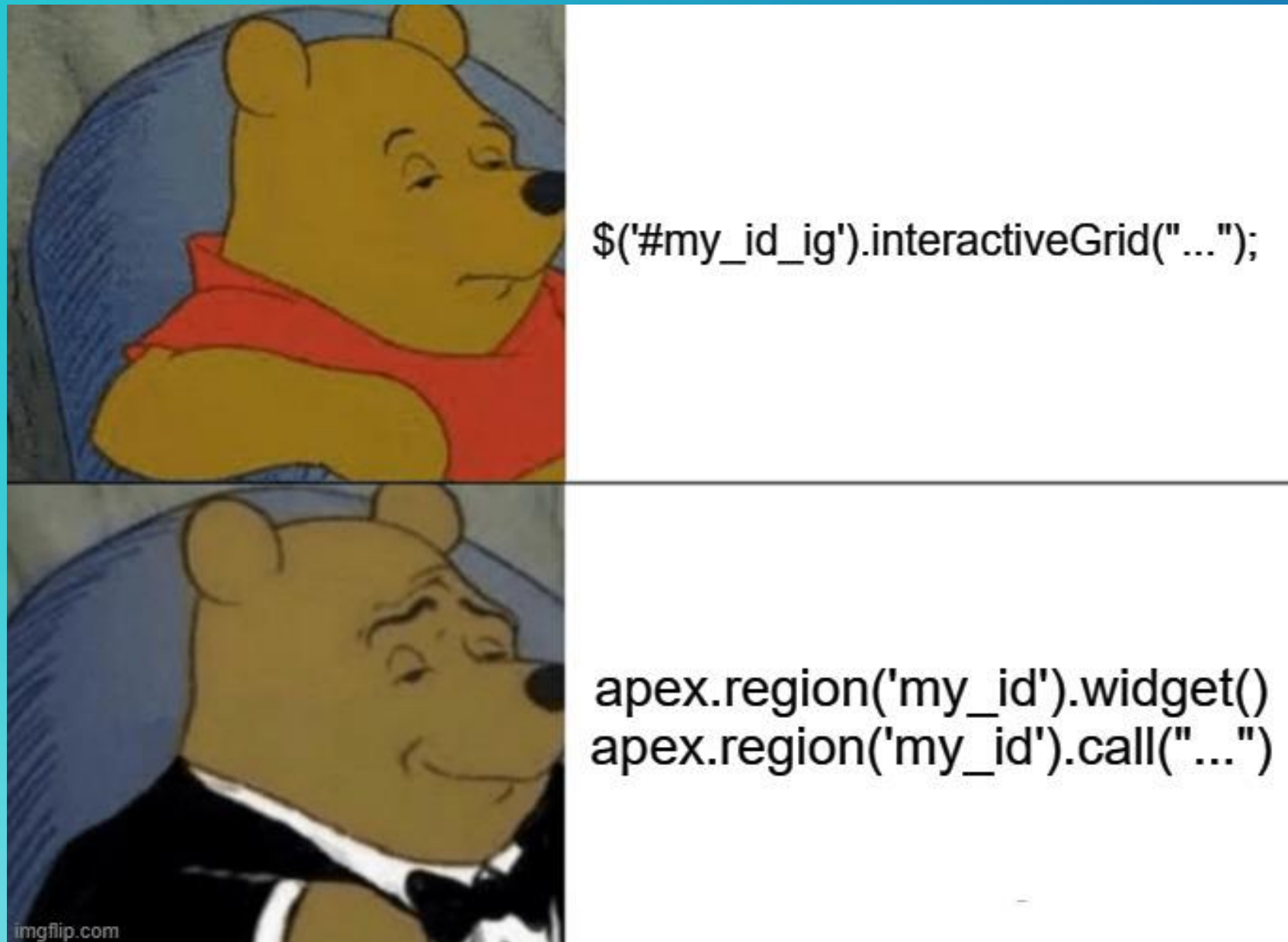


```
$('#R51572603115080_ig').interactiveGrid("getViews");
```

```
$('#my_meaningful_id_ig').interactiveGrid("getViews");
```



APEX REGION INTERFACE



`apex.region("my_id")` → Our IG Region

`.widget()`
`.call()` → Region Interface methods

`apex.region("my_id").widget()`
→ `$("#my_id_ig")`

`apex.region("my_id").call("...")`
→ `$("#my_id_ig").interactiveGrid("...")`

Widgets

grid
iconList
interactiveGrid
menu
menuButton
recordView
tableModelView
tableModelViewBase
treeView

Namespaces

apex
apex.actions
apex.da
apex.debug
apex.event
apex.item
apex.region

Interfaces

actions
htmlBuilder
interactiveGridView
item
model
region
treeNodeAdapter

Widget: interactiveGrid

QuickNav

Options

- defaultDetailViewOptions
- defaultGridViewOptions
- defaultIconViewOptions
- defaultModelOptions
- defaultSingleRowOptions
- editable
- features
- initActions
- initialSelection
- reportSettingsArea
- saveLoadingIndicator
- saveLoadingIndicatorPosition
- toolbar
- toolbarData
- trackParentSelection

Events

- modechange
- reportchange
- reportsettingschange
- save
- selectionchange
- viewchange
- viewmodelcreate

Methods

- copyDefaultToolbar
- focus
- getActions
- getCurrentView
- getCurrentViewId
- getSelectedRecords
- getToolbar
- getViews
- gotoCell
- refresh
- resize
- setMasterRecord
- setSelectedRecords

Type Definitions

- toolbarData

interactiveGrid

apex
apex.actions
apex.da
apex.debug
apex.event
apex.item
apex.lang
apex.locale
apex.message
apex.model
apex.navigation
apex.navigation.dialog
apex.navigation.popup
apex.page
apex.region
apex.server
apex.storage
apex.theme
apex.util
apex.util.delayLinger
apex.widget

Interfaces

actions
htmlBuilder
interactiveGridView
item
model

Overview

A jQuery UI widget for the Interactive Grid component of Application Express. It is a modern component combining powerful reporting features, with easy multi-row editing. It is exposed as a native region type in APEX, and has a number of declarative features available for customization in Page Designer. In addition to what is declaratively available, there is tremendous scope for performing further JavaScript customizations, to achieve more specific functionality.

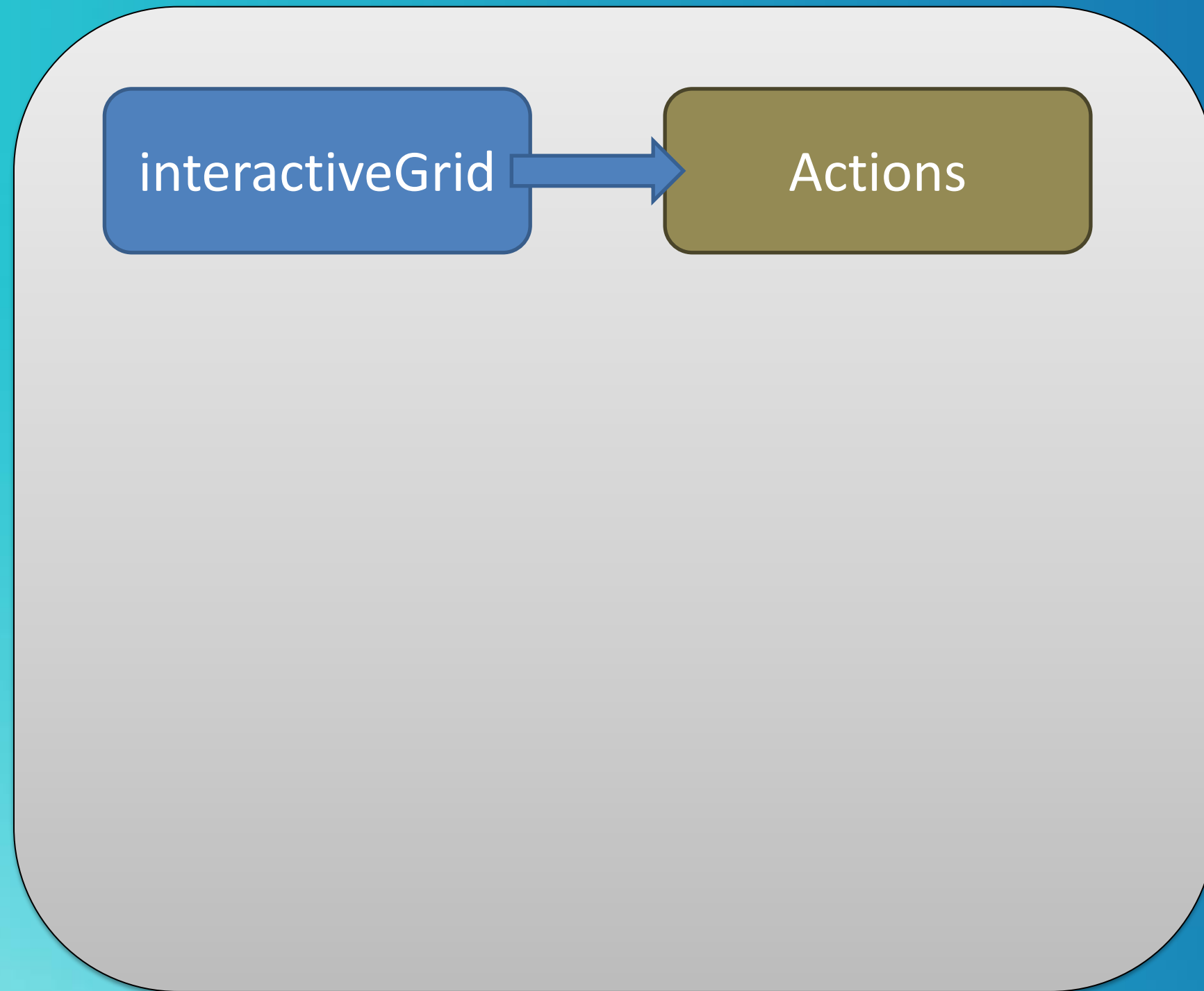
This widget is used as the main interface with APEX, and is comprised of a number of other widgets and making use of a number of other JavaScript modules including its own data model layer, which are important to understand when performing customizations. The Interactive Grid widget uses the following:

- [grid widget](#)
- [recordView widget](#)
- [tableModelView widget](#)
- [tableModelViewBase widget](#)
- toolbar widget (see `widget.toolbar.js` file for further information)
- [menu widget](#)
- [iconList widget](#)
- [apex.actions module](#)
- [apex.model module](#)
- [apex.item module](#)

See each for further details, although the interactiveGrid widget documentation will make numerous mention of them to demonstrate certain concepts.

The intention is not to enable developers to create their own Interactive Grid instances, as this would create great burden in developing supporting functionality, for example the model layer. Rather, that developers can

APEX ACTIONS

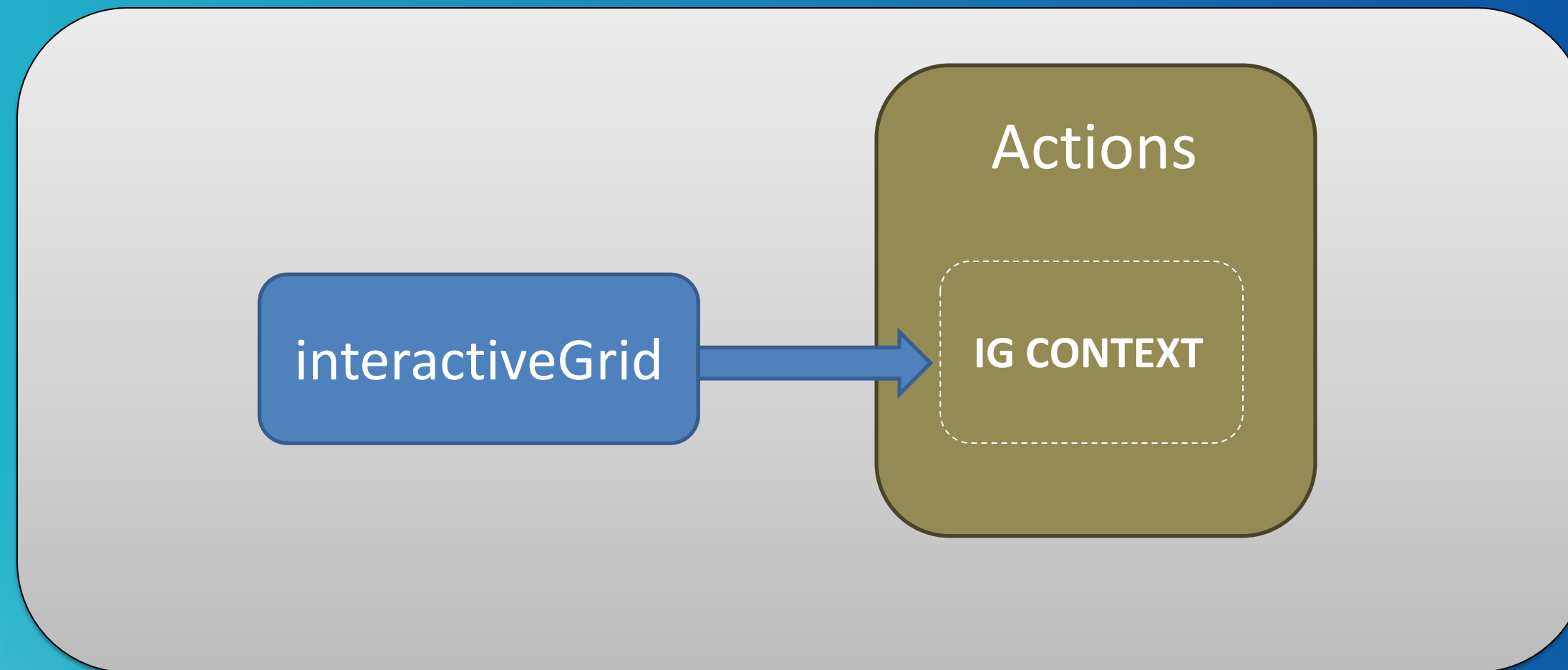


```
{  
  "name"      : [string],  
  "label"     : [string],  
  "icon"      : [string],  
  "action"    : [function],  
  "get"       : [function],  
  "set"       : [function],  
  "disable"  : [boolean],  
  ...  
}
```

Action types:

- Action
- Toogle
- Radio Group

ACTIONS CONTEXT



```
> myGridActions = apex.region('my_interactiveGridId').call("getActions");
```

Interface: actions

QuickNav

Properties

- context
- typeName

Methods

- add
- addFromMarkup
- addShortcut
- clear
- disable
- disableShortcuts
- enable
- enableShortcuts
- get
- hide
- invoke
- list
- listShortcuts
- lookup
- observe
- remove
- removeShortcut
- set
- shortcutDisplay
- show
- toggle
- unobserve
- update
- updateChoices



Type Definitions

- action
- shortcutListItem
- shortcutName

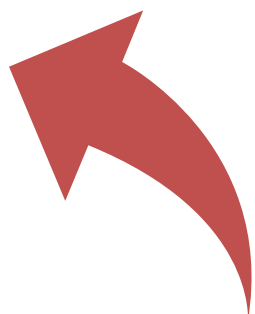
Index

Widgets

- grid
- iconList
- interactiveGrid**
- menu
- menuButton
- recordView
- tableModelView
- tableModelViewBase
- treeView

Namespaces

- apex
- apex.actions
- apex.da
- apex.debug
- apex.event
- apex.item
- apex.lang
- apex.locale
- apex.message
- apex.model
- apex.navigation
- apex.navigation.dialog



Pre-defined actions used by the Interactive Grid widget

Name	Type	Description
search	Action	Handles the main search functionality available from the search field in the toolbar.
search-case-sensitive	Toggle	Toggle whether the search is case-sensitive.
filter-column	Radio	Constrain search to a specific column.
change-report	Radio	Switch to a different saved report.
change-view	Action	Switch to a different view (eg grid, icon, detail, etc.).
chart-view	Action	Switch to the chart view. Note: This is handled differently to other views because this is user-defined and may not be configured.
show-columns-dialog	Action	Show the columns dialog.
show-filter-dialog	Action	Show the filter dialog.
show-sort-dialog	Action	Show the sort dialog.
show-aggregate-dialog	Action	Show the aggregate dialog.
show-	Action	Show the flashback dialog.

USE CASES



```
1 apex.region('my_gird_id').call('getActions').invoke('save');
```



```
1 apex.region('my_gird_id').call('getActions').invoke('show-columns-dialog');  
2 apex.region('my_gird_id').call('getActions').invoke('show-filter-dialog');  
3 apex.region('my_gird_id').call('getActions').invoke('show-sort-dialog');  
4 apex.region('my_gird_id').call('getActions').invoke('show-aggregate-dialog');  
5 apex.region('my_gird_id').call('getActions').invoke('show-flashback-dialog');  
6 apex.region('my_gird_id').call('getActions').invoke('show-highlight-dialog');  
7 apex.region('my_gird_id').call('getActions').invoke('show-control-break-dialog');  
8 apex.region('my_gird_id').call('getActions').invoke('show-download-dialog');  
9 apex.region('my_gird_id').call('getActions').invoke('show-help-dialog');
```

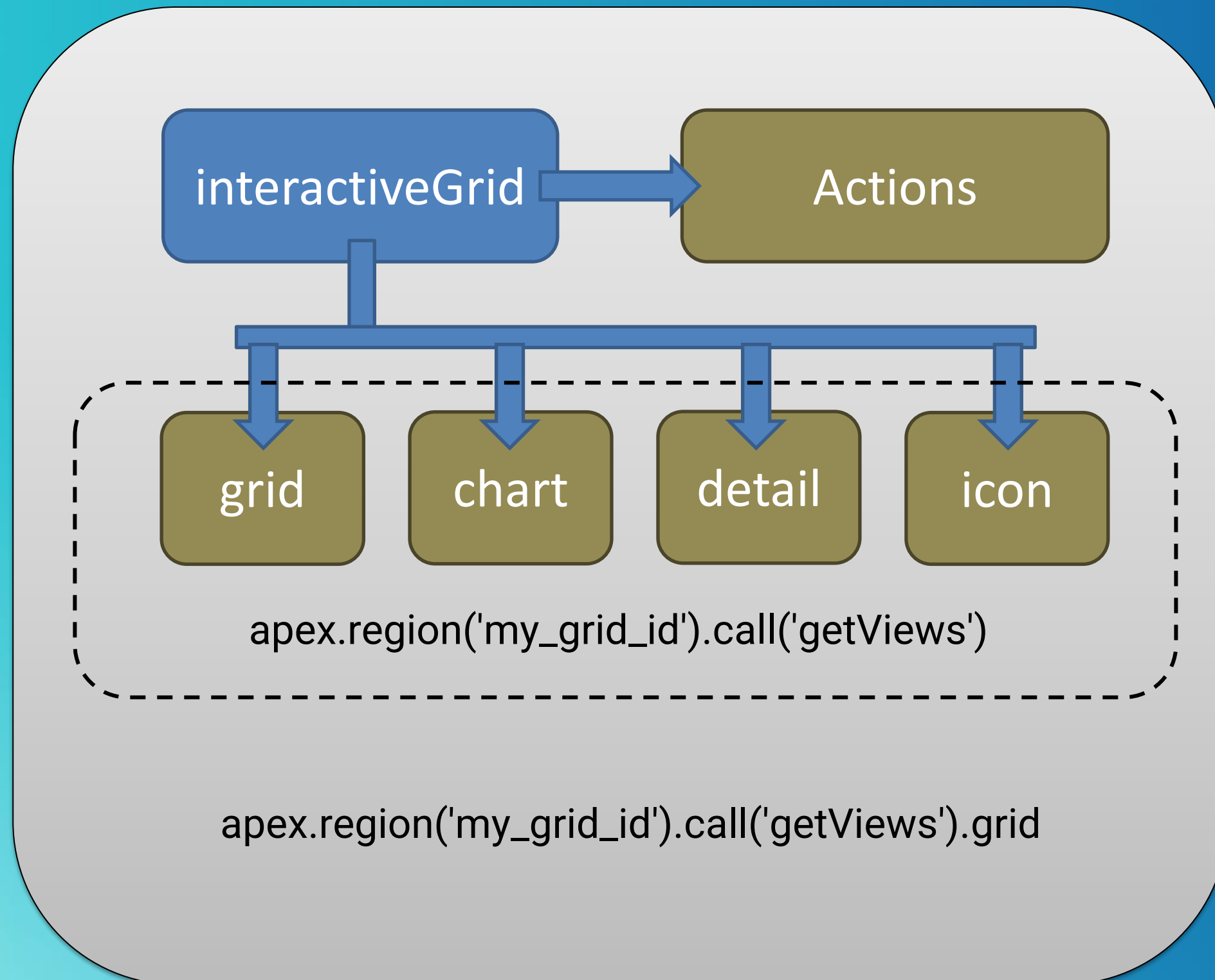


```
1 apex.region('my_gird_id').call('getActions').toggle('edit');
```





INTERACTIVE GRID VIEWS



Methods

- copyDefaultToolbar
- focus
- getActions
- getCurrentView
- getCurrentViewId
- getSelectedRecords
- getToolbar
- getViews
- gotoCell
- refresh
- resize
- setMasterRecord
- setSelectedRecords

Interface: interactiveGridView

QuickNav

Properties

- [cssClass](#)
- [icon](#)
- [internalIdentifier](#)
- [model](#)
- [modelName](#)
- [singleRowMode](#)
- [singleRowView\\$](#)
- [title](#)
- [view\\$](#)

Methods

- [getContextRecord](#)
- [getSelectedRecords](#)
- [setSelectedRecords](#)

interactiveGridView

Defines an Interactive Grid view. The [interactiveGrid](#) widget manages a number of different views of the report data. Each view is implemented by a [interactiveGridView](#) interface. The types of views are:

- [grid](#): This is implemented by the [grid](#) widget. It also supports single row view.
- [chart](#): This is implemented by JET [ojchart](#).
- [detail](#): This is implemented by the [tableModelView](#) widget.
- [icon](#): This is implemented by the [tableModelView](#) widget. The main difference between this and detail view is that this supports selection. The [tableModelView#useIconList](#) option is true.

Each view has a [model](#) to manage its data and a [widget](#) to implement the user interface. Most of the properties and methods of the view are for internal use by the [interactiveGrid](#) widget. A few of them are useful in advanced use cases to control or configure interactive grid.

WORTH TO REMEMBER

Properties

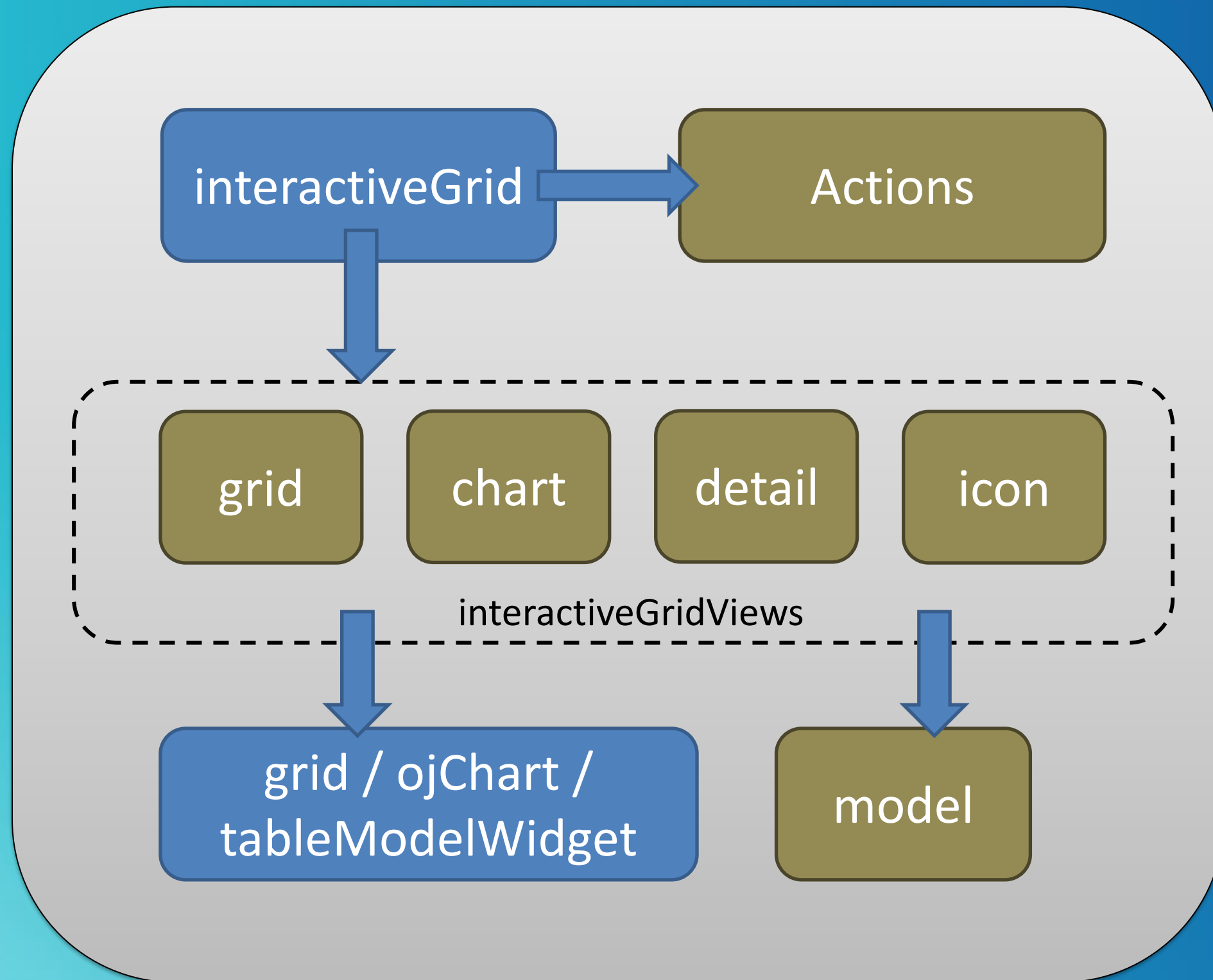
- `cssClass`
- `icon`
- `internalIdentifier`
- `model`

- `modelName`
- `singleRowMode`
- `singleRowView$`
- `title`
- `view$`

Methods

- `getContextRecord`
- `getSelectedRecords`
- `setSelectedRecords`

COMPLETE PICTURE



Legend

Interfaces

jQuery UI
Widgets



DEMO

OTHER PRETIUS PRESENTATIONS



DOMINIK GRABIŃSKI

Talk to Me Gently: How to Integrate Oracle DB with Google Assistant and Add It to the APEX Application



RAFAŁ GRZEGORCZYK

DevOps your Oracle Database with Liquibase in less than 1 hour

Powered by

pretius

www.pretius.com

Adam Kierzkowski

akierzkowski@pretius.com

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

 [akierzkowski](https://github.com/akierzkowski)

Find more:

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

 facebook.com/pretius