Sencha

Amazing Web Apps for Every Device

ExtJS 4.1 Performance

Nigel White, Software Architect

Examples & slides

- https://github.com/ExtAnimal/JSConf2012
- https://github.com/ExtAnimal/JSConf2012.git

Create a Panel

```
new Ext.panel.Panel({
    title: ‘MyPanel’,
    height: 400,
    width: 600,
    tbar: [{
        text: ‘A Button’
    }],
    renderTo: document.body
});
```

http://jsfiddle.net/ExtAnimal/Wap3y/6/

A Simple Grid View

- A View displays entities.
- Entity properties are formatted according to a template.
- Grids define a template which encapsulates entity properties in table cells.

Define an Entity

```
Ext.define('Employee', {
    extend: 'Ext.data.Model',
    fields: [
        {name: 'rating', type: 'int'},
        {name: 'salary', type: 'float'},
        {name: 'name'}
    ]
});
```

Create a Store

```
var store = Ext.create('Ext.data.Store', {
    id: 'store',
    data: createFakeData(500),
    model: 'Employee',
    proxy: {
        type: 'memory',
        reader: 'json'
    }
});
```

- A Proxy communicates with a separate data source
- `'memory'/'ajax'/'localstorage'`
- A Reader creates entities from raw data from the Proxy
- `'json'/'xml'`

Define Columns

```
{
    text: 'Name',
    flex: 1,
    sortable: true,
    dataIndex: 'name'
}
```

- Defines a grid column
- Specify `width` or `flex`
- `dataIndex` references a field name from the Model

A Grid is a Panel subclass

```
new Ext.grid.Panel({
    // Other configs are the same
    // Some extra configs for grids:
    store: myStore,
    columns: [{...}],
    selModel: {}
});
```

examples/jsconf/jsconf-grid.html

500 Employees

Apply Pay Rise slowly Apply Pay Rise quickly

	Name	Rating	Salary
1	David Robinson	4	$1,500.00
2	Dave Ferrero	3	$900.00
3	Dave Spencer	1	$400.00
4	Abe White	3	$100.00
5	Nicolas Maintz	3	$400.00
6	Abe Davis	4	$100.00
7	Adam Ferrero	1	$900.00
8	Aaron Kaneda	2	$900.00
9	Tommy Conran	4	$100.00
10	Dave Avins	2	$1,500.00

Interaction

```
{
    xtype: 'actioncolumn',
    width: 30,
    sortable: false,
    bubbleEvents: 'rowDelete',
    items: [{
        icon: '../shared/icons/fam/delete.gif',
        tooltip: 'Delete Plant',
        handler: function(grid, rowIdx, colIdx) {
            this.fireEvent('rowDelete', rowIdx);
        }
    }
}
```

Panel listens for bubbled event

```
listeners: {
    rowDelete: function(rowIndex) {
        store.removeAt(rowIndex);
    }
}
```

Network Latency

CSS processing

Javascript execution

DOM

What slows your app down?

Network Latency

- Use Sencha Command
- http://docs.sencha.com/ext-js/4-1/#/guide/command
- http://www.sencha.com/blog/all-new-sencha-cmd/
- Concatenate all **required** Javascript into `all-classes.js`
- sencha compile -classpath=path/to/src page -in array-grid.html -out build/index.html

CSS processing

- Avoid inefficient, over-specific selectors
- Selectors are matched right to left
- `.HeaderContainer .nav span`

Javascript execution

Javascript execution

- Avoid older and badly written JS engines. ;)
- Optimize code which is repeated.
- Optimize any code executed at render or *initial* layout time in an ExtJS app
- Better still, do not perform extra processing at render time.

Javascript execution

- Move invariant expressions out of loops
- Use for (...) {} rather than Ext.Array.each
- If a function only performs its functionality under some condition, test that condition ***outside*** *the call*.
- Setup and teardown of a call frame is slow on bad JS engines.

Array Grid

Price ▾ Change

Get total in bad way

Get total in good way

	Price	Change	% Change	Last Updated
	$71.72	0.02	0.03%	09/01/2012
	$29.01	0.42	1.47%	09/01/2012
Altria Group Inc	$83.81	0.28	0.34%	09/01/2012
American Express Company	$52.55	0.01	0.02%	09/01/2012
American International Group, Inc.	$64.13	0.31	0.49%	09/01/2012
AT&T Inc.	$31.61	-0.48	-1.54%	09/01/2012
Boeing Co.	$75.43	0.53	0.71%	09/01/2012
Caterpillar Inc.	$67.27	0.92	1.39%	09/01/2012
Citigroup, Inc.	$49.37	0.02	0.04%	09/01/2012
E.I. du Pont de Nemours and Company	$40.48	0.51	1.28%	09/01/2012
Exxon Mobil Corp	$68.10	-0.43	-0.64%	09/01/2012
General Electric Company	$34.14	-0.08	-0.23%	09/01/2012
General Motors Corporation	$30.27	1.09	3.74%	09/01/2012

A Menu of operations per column

Github

examples/jsconf/

```
sencha compile
    -classpath=../../src page
    -in array-grid.html
    -out build/index.html
```

```
tbar: [{
    text: 'Price',
    field: 'price', // Button configured with column
    menu: [{
        text: 'Get total in bad way',
        handler: badTotalFn
    }, {
        text: 'Get total in good way',
        handler: goodTotalFn
    }]
}, {
    text: 'Change',
    menu: []
}]
```

Column’s Button knows the field

```
function badTotalFn(menuItem) {
   var r = store.getRange(),
       total = 0;

   Ext.Array.each(r, function(rec) {
       total += rec.get(menuItem.up('button').field);
   });
}
```

What’s wrong with this?

```
function goodTotalFn(menuItem) {
    var r = store.getRange(),
        field = menuItem.up('button').field;
        total = 0;

    for (var j = 0, l = r.length; j < l; j++) {
        total += r[j].get(field);
    }
}
```

Why is this better?

Browser	Bad	Good
Chrome	1700ms	10ms
IE9	18000ms	500ms
IE6	Gave up	532ms

The results (10000 iterations)

Use PageAnalyzer

- extjs/examples/page-analyzer/page-analyzer.html
- Analyzes method calls of selected classes
- Start Chrome with --enable-benchmarking for microsecond time measurement accuracy.

Performance Accumulators

```
{
    "Component.up": {
        "Ext.Component": "up"
    },
    "CQ.is": {
        "Ext.ComponentQuery": "!is"
    }
}
```

Data

Name	Time	Time/Call	Samples	Min Calls	Max Calls
Path: /~nigewhite/Sencha/JSConf2012/extjs/examples/jsconf/build/ (x - 1)					
Component.up	0	0	1	29	29
CQ.is	0	0	1	65	65
Path: /~nigewhite/Sencha/JSConf2012/extjs/examples/jsconf/build/ (x - 2)					
Component.up	0	0	1	1	1
CQ.is	0	0	1	9	9

Results (bad way first)

Update Views Efficiently

- Views (grids and trees) up updated in response to store events
- “add”, “remove” and “update” events all update attached views
- When modifying multiple records, suspend store events

```
{
    myGrid.store.suspendEvents();
    // update many records
    myGrid.store.resumeEvents();
    myGrid.view.refresh();
}
```

Trees are grids too

- Trees are the same as grids.
- The view has a store which is a flattened "view" of the nodes

```
{
    myTree.view.store.suspendEvents();
    // update many records
    myTree.view.store.resumeEvents();
    myTree.view.refresh();
}
```

Coalesce multiple layouts

- ExtJS 4.x performs a layout upon content change or size change.
- Coalesce multiple changes into one layout run

```
{
    Ext.suspendLayouts();
    // batch of updates
    Ext.resumeLayouts(true);
}
```

Coalesce multiple layouts

- ExtJS 4.x performs a layout upon content change or size change.
- Coalesce multiple changes into one layout run

```
{
    Ext.suspendLayouts();
    // batch of updates
    Ext.resumeLayouts(true);
}
```

Reduce DOM burden

- Reduce Container nesting
- Use the simplest Container or Layout which does the job

```
{
    xtype: 'tabpanel',
    items: [{
        title: 'Results',
        items: {
            xtype: 'grid'
            ...
        }
    }]
}
```

What's wrong with this?

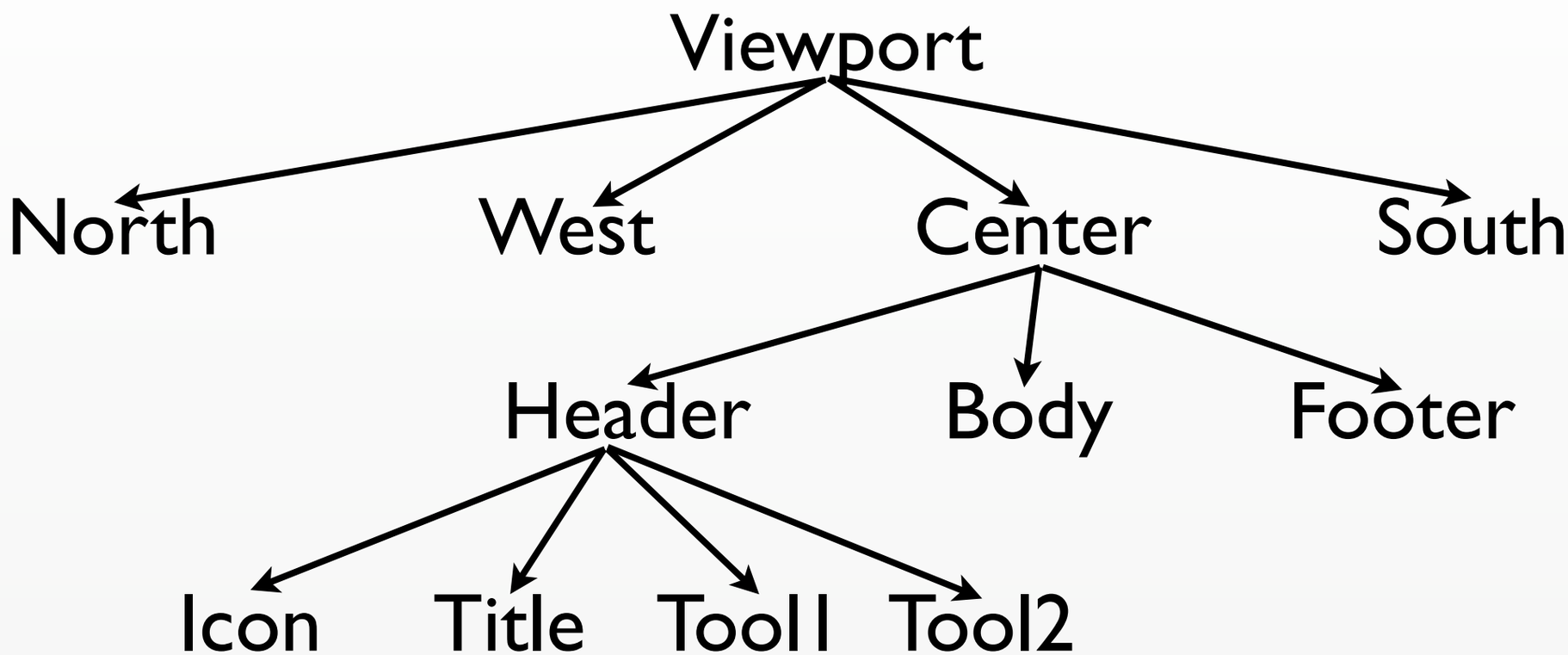
```
{
    xtype: 'tabpanel',
    items: [{
        title: 'Results',
        xtype: 'grid',
        ...
    }]
}
```

Why is this better?

Why is this important?

- In ExtJS 4.x, more widgets are Containers which must lay out child Components using a Layout Manager. eg: Header, TabBar.
- In ExtJS 4.x, All Components lay out their internal DOM structure using a Component Layout Manager (Where 3.x simply had an onResize method)

The Component Tree



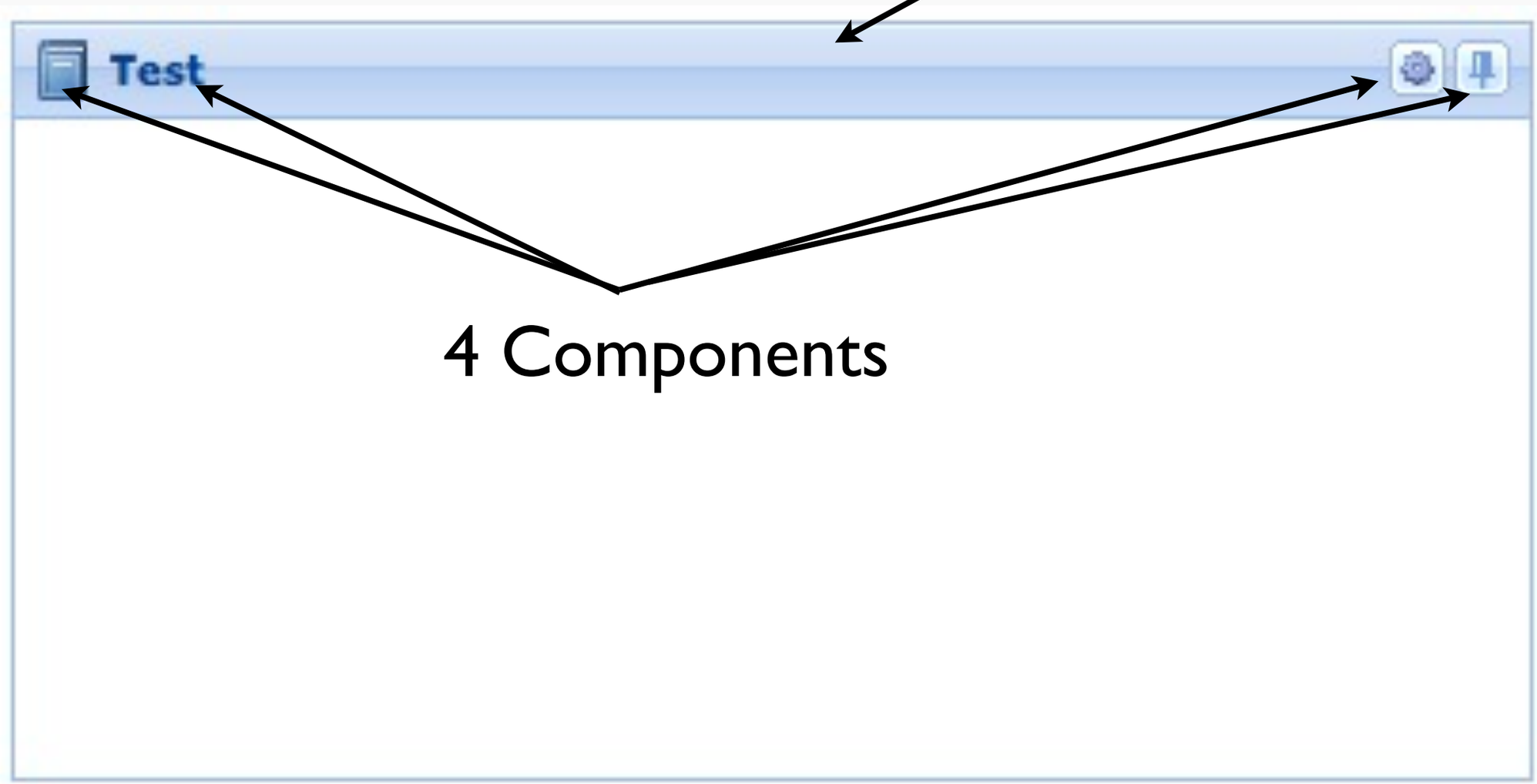
- Walked twice, visiting **every** node.
- beforeRender & getRenderTree: Finished pre-render configs, Contributes a DomHelper config which is converted to markup and added into the buffer.
- onRender & afterRender: Inform the Component about its element. Component grabs child element references (childEls)

Example Panel

```
Ext.create('Ext.panel.Panel', {
    width: 400, height: 200,
    icon: '../shared/icons/fam/book.png',
    title: 'Test',
    tools: [{
        type: 'gear'
    }, {
        type: 'pin'
    }],
    renderTo: document.body
});
```

Example Panel

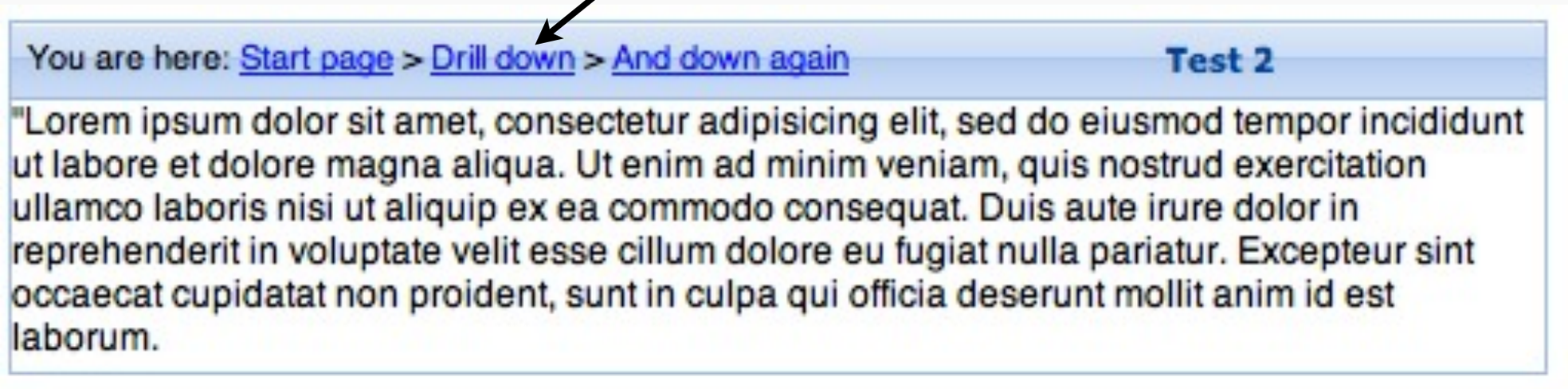
Header = Container + HBox



4 Components

More flexibility than ExtJS 3.x

Components in Headers:
"Breadcrumbs" DataView



"shrinkwrap" sizing where content drives height.
(and/or width for non-textual content)

Github

- examples/panel/panel-test1.html
- examples/panel/panel-test2.htm

Layout analysis, auto sized.

Layouts

Layout	Box Parent	Time	Calls	Avg Time	Tot Time
Run #9 (2012-04-23 16:1:		15	1	15	15
Run #10 (2012-04-23 16::		47	1	47	47

More tries to lay out

Layout	Time	Calls	Avg Time	Tot Time
Run #1 (2012-04-27 12:16:16)	16.48	1	16.48	16.48
Flush	1.8	7	0.26	1.8
Invalidate	5.59	1	5.59	5.59
CompleteLayout	0.2	2	0.1	0.2
FinalizeLayout	0.01	1	0.01	0.01
FinishedLayout	0.31	1	0.31	0.31
AfterLayout	0.53	1	0.53	0.53
panel<dock>	2.21	3	0.74	7.29
panel<autocontainer>	0.34	1	0.34	0.34

It took **3** tries to do the dock layout.
Because it has to be done in stages.

Auto sized layout

- Two layout runs due to deferInitialRefresh default as true
- More expensive layout after the data is refreshed
- The heights of all Components in header must be read from the DOM
- That requires that the width be set - because width **may** affect height of Components (wrapping)

Layout analysis, fixed size

Layouts

Layout	Box Parent	Time	Calls	Avg Time	Tot Time
Run #6 (2012-04-23 16:1(		15	1	15	15

- One layout run.
- Faster because the height was known before any layout was performed, so docking could proceed immediately.

Waves of change

- Write to DOM then read from DOM == a browser reflow. Do as few as possible.
- Layouts all executed - *publish* all the information they can using what is already *published* by other layouts.
- If no layout has made progress, then all *published* values are *flushed* to the DOM.
- Layouts begin another cycle - they may read the results from that flush.

"publish" means the value is available to other layouts. These values are *not* yet in the DOM

Dock Layout: First try

- Width is already *published* - configured width.
- Not height - Panel is shrinkwrapping content.
- Try to dock the header - *publish* its width.
- *Publish* body x position and width.
- Header height not known, (internal components have not performed layouts and published sizes) so no further progress can be made on the dock.

Second try

- Header has published its height.
- We can therefore position the body - y = 25px
- But the body height cannot be measured because no flush has taken place to set the body element width.
- No further progress can be made on finishing the dock layout with the total panel height

First DOM flush takes place

Third try

- Body's width has now been flushed to the DOM, so its height can be measured.
- Header already published its height.
- The total height of the Panel can now be calculated
- Done!
- But maybe another layout was waiting on *this* one!

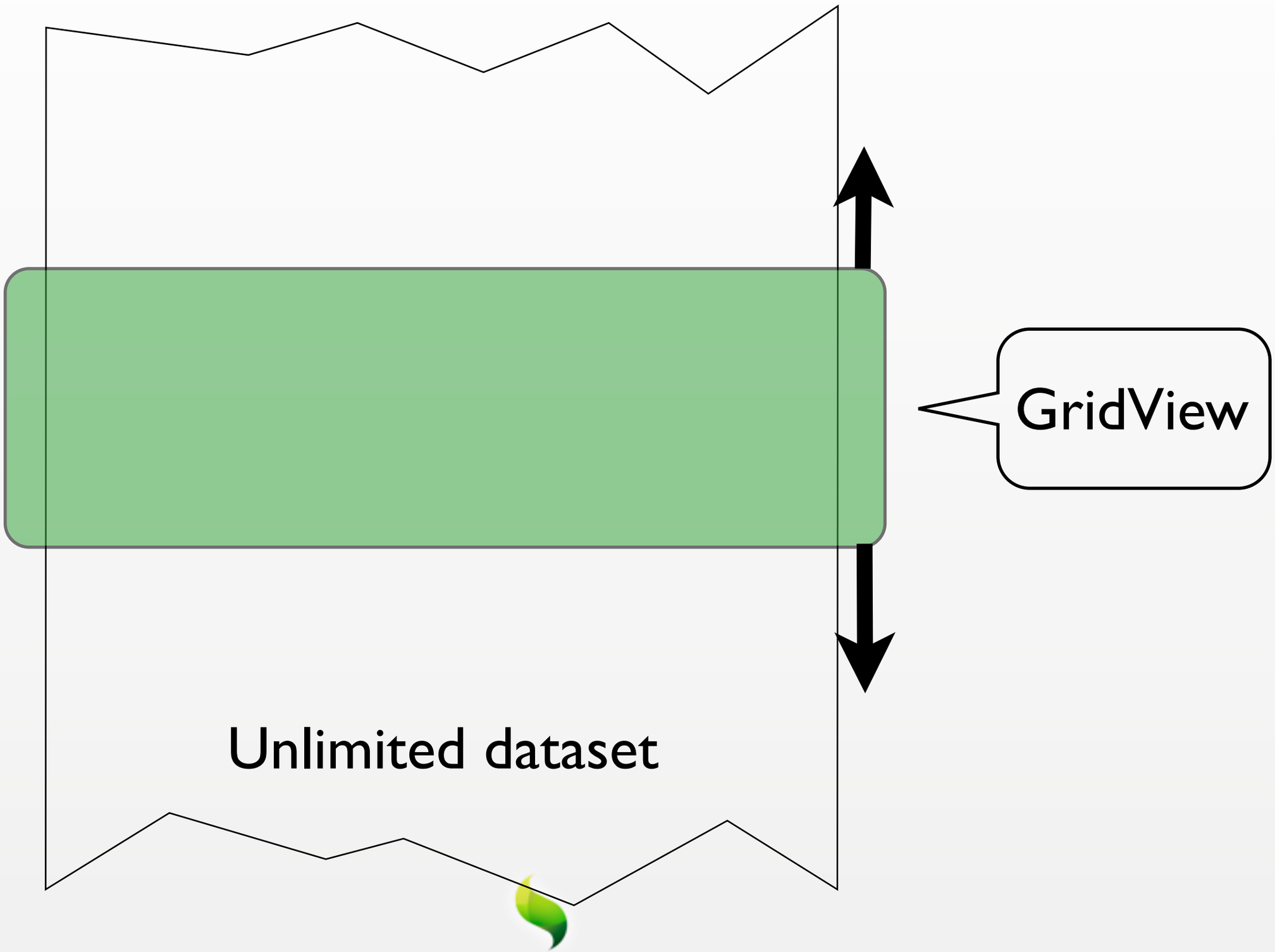
TL;DL

- Avoid “auto” sizing (shrinkwrapping) in the initial page layout. It will require multiple passes to flush the sizes, then measure.
- Box layout’s stretchmax config causes multiple layout attempts.
- Avoid size constraints (minHeight, maxWidth etc). If they kick in, they will cause another round of layout calculations.

Grid Performance

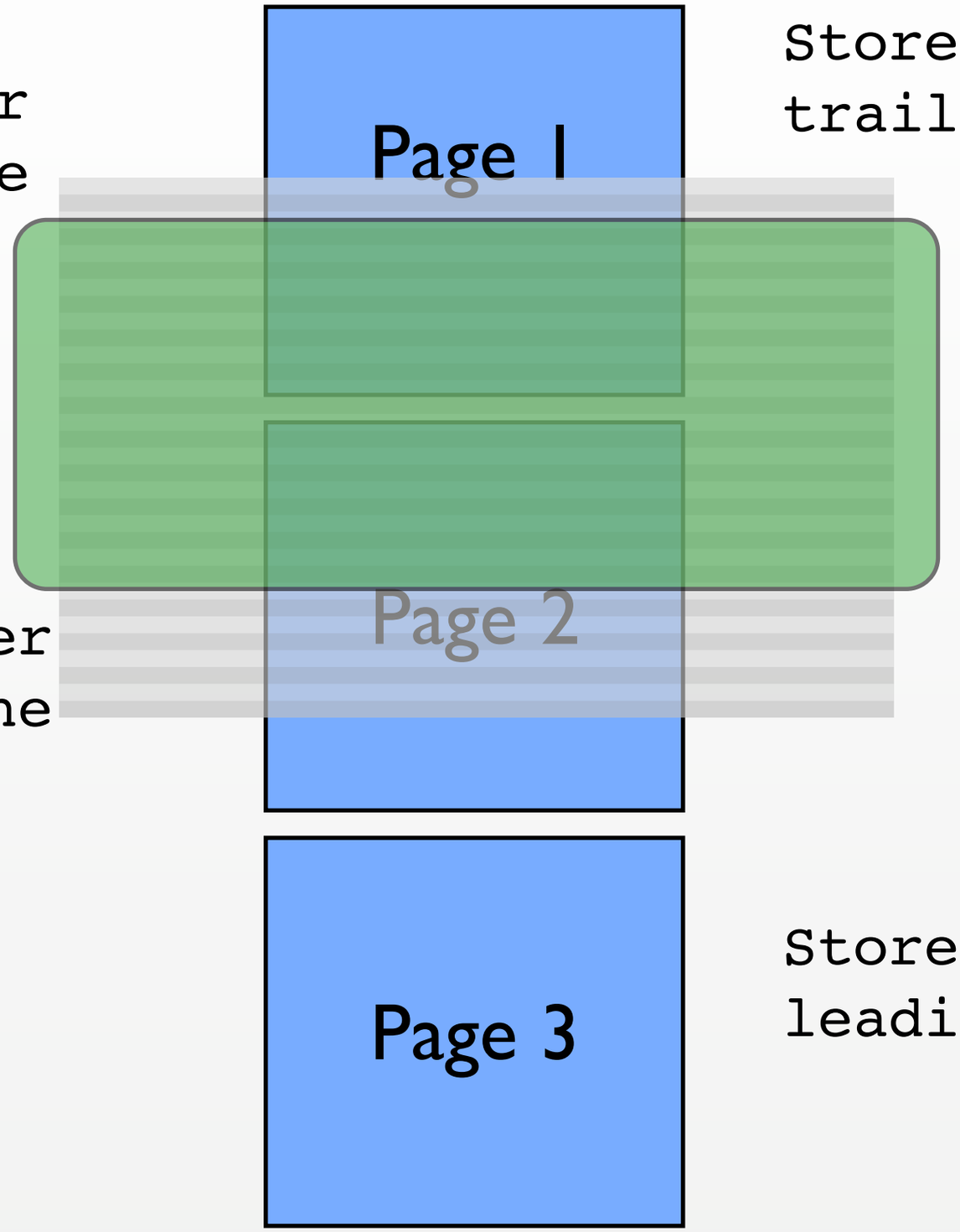
- Table size affects performance
- Column count has a large effect.
- Avoid trackMouseOver on slow browsers
- Where dataset is large, use *buffered rendering*.

Letterbox view



PagingScroller
trailingBufferZone

Store
trailingBufferZone



PagingScroller
leadingBufferZone

Store
leadingBufferZone

Questions?

- https://github.com/ExtAnimal/SourceConf2012
- git@github.com:ExtAnimal/SourceConf2012.git