

Adding a Gantt chart to your Ext JS application

Mats Bryntse, CEO @Bryntum







- Mats Bryntse, from Stockholm Sweden 🇸🇪
- First saw Ext JS in 2007, fell in love with the grid ❤️
- In 2009, I started Bryntum
- We make UI components and dev tools for web dev

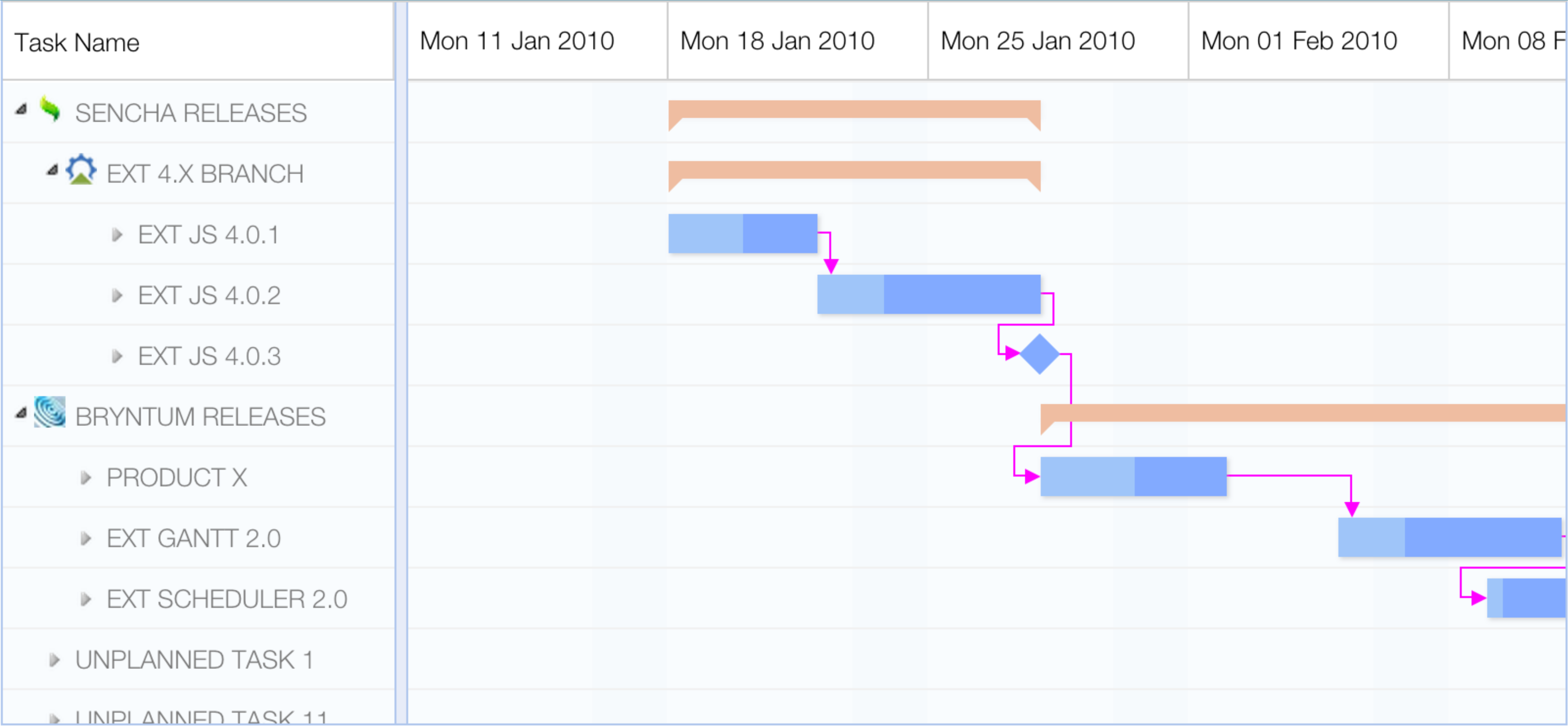
Scheduler for Ext JS

Tab 1 - Some other component

Tab 2 - Scheduler


Staff		Time all...	w.48 Dec 2010					
			1	2	3	4	5	6
	Arnold R&D	10.0 d		Paris Trip				
	Lisa CEO	5.0 d				Board Meeting		
	Dave Acceptance Test	8.0 d	Test IE8			Test IE8 Some More		
	Lee Sales	8.0 d	Conference X				Meet Client	

Gantt for Ext JS



Task Board for Ext JS

||| Not Started (2) <<



Fix IE7 bug

Sneak-install Chrome Frame

Add new task...

||| In Progress (3) <<

Add Windows Phone support

IE6 support

Make App


||| Test (1) <<

Find Unicorn

||| Acceptance (3)

||| Done (4) <<

Change floor tiles



Chrome development

Dig hole

Find holy grail

Siesta - Testing tool for any JS / web

Filter tests

Pass Fail

Run test

grid/examples/filtering.t.js

Siesta

Using ES 6 modules

Docs

Common

Grid

Column

Examples

examplebro 0 0

basic.t.js 0 0

bigdataset.t 0 0

celledit.t.js 0 0

columns.t.js 0 0

columndrag 0 0

columntype. 0 0

contextmen 0 0

features.t.js 0 0

filtering.t.js 20 0

filterbar.t.js 0 0

groupedhea 0 0

grouping.t.js 0 0

groupsummr 0 0

Sanity

Waited 0 ms for Grid is here

Grid is here

Context menu feature is here

Filtering feature is here

Initial filter (city) set on store

Records filtered

Filter icon on header

Clicking Remove all filters

All filters removed from store

All records available

Filter removed from header

Context menu, string

Filter (city) set on store

Records filtered

Filter icon on header

Context menu, number

Filter (age) set on store

Records filtered

Filter icon on header

record deletion while grouped and filtered

No "NaN" string found in DOM

Filtering demo

#	Name	Age	City
Washington			
1	Doug M Thomas	36	Washingtc
2	Adam P Johnson	80	Washingtc
3	Jenny R Williams	81	Washingtc
4	Karen A Ewans	79	Washingtc
5	Don P Thomas	82	Washingtc
6	Jane F Taylor	61	Washingtc
7	Lisa J More	80	Washingtc
8	David L McGregor	47	Washingtc

20

0

✓

✗

Ext JS class system

- Extending a class
- Component model
- Life cycle
- Template methods

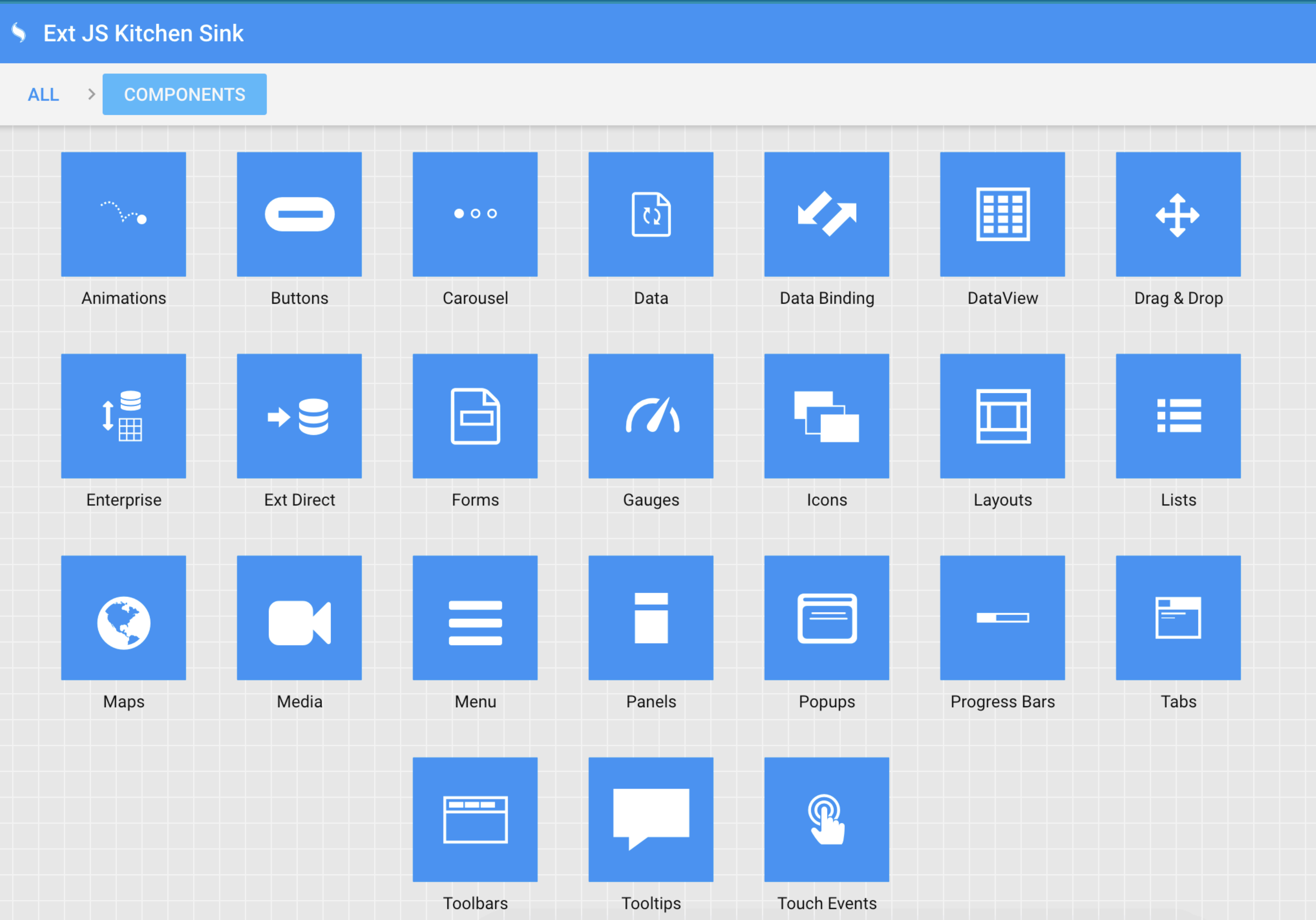
Ext Gantt

- Intro
- Architecture
- Scheduling Engine
- Adding to an Ext application

Hands up: Who is using Ext JS?

One of the great strengths of Ext JS:
Mature **reusable** UI components.

Ext JS UI components





Ext JS Class System

Introduction

- Legacy JS lacked real classes (added in ES6)
- Ext JS has its own class system
- Dynamic loading
- Classes defined as a String + Object with properties

Before a class is defined, all its dependencies are loaded

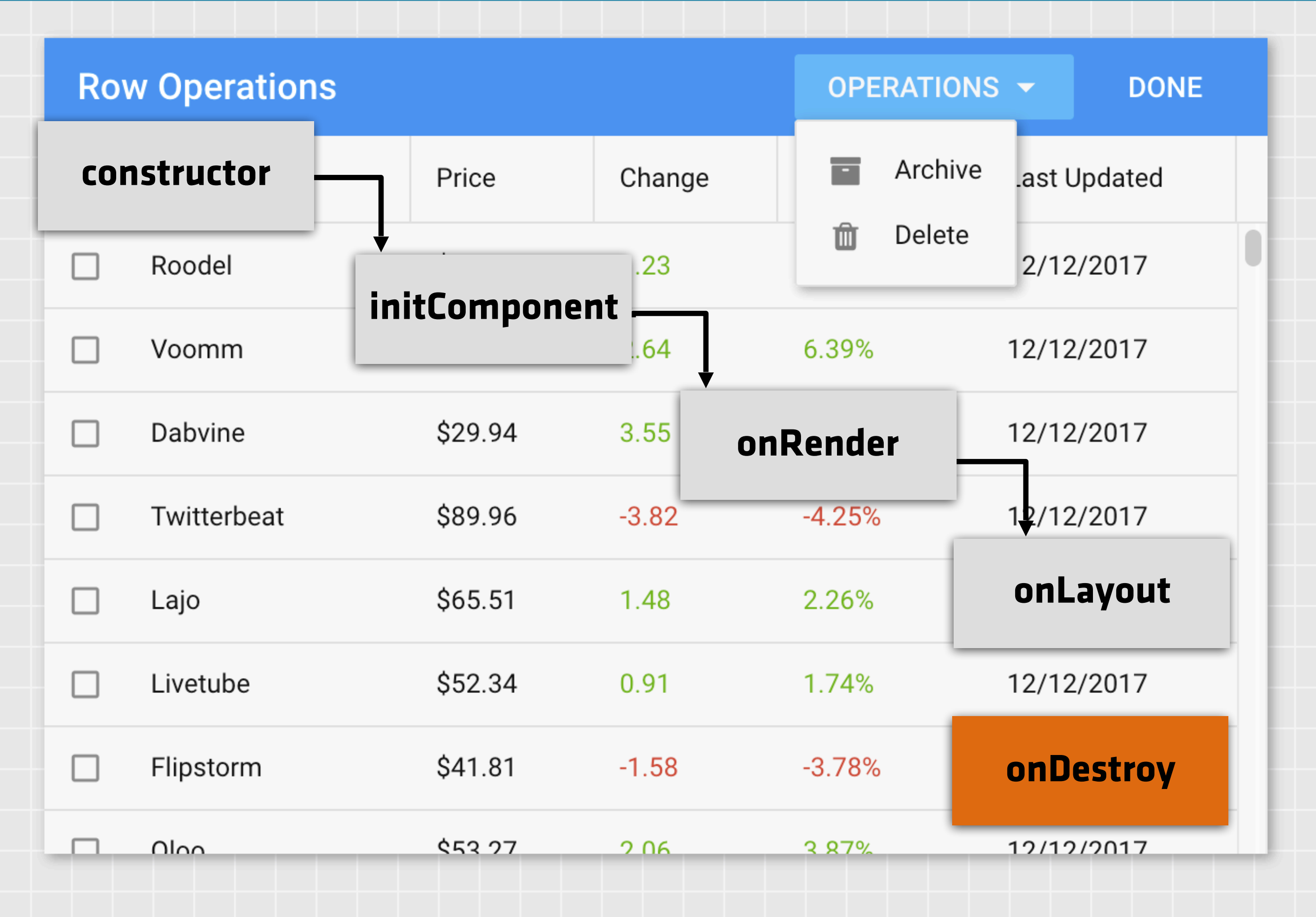
```
Ext.define("MyApp.FilterField", {  
    extend      : "Ext.form.TextField",  
    requires    : ["Ext.other.Class"],  
  
    width       : 150,  
    ...  
});
```



Everything visual in Ext JS is a **Component**

Row Operations				OPERATIONS ▼	DONE
<input type="checkbox"/>	Company	Price	Change	<div><div>Archive</div><div>Delete</div></div>	Last Updated
<input type="checkbox"/>	Roodel	\$59.47	1.23		2/12/2017
<input type="checkbox"/>	Voomm	\$41.31	2.64	6.39%	12/12/2017
<input type="checkbox"/>	Dabvine	\$29.94	3.55	11.86%	12/12/2017
<input type="checkbox"/>	Twitterbeat	\$89.96	-3.82	-4.25%	12/12/2017
<input type="checkbox"/>	Lajo	\$65.51	1.48	2.26%	12/12/2017
<input type="checkbox"/>	Livetube	\$52.34	0.91	1.74%	12/12/2017
<input type="checkbox"/>	Flipstorm	\$41.81	-1.58	-3.78%	12/12/2017
<input type="checkbox"/>	Oloo	\$53.27	2.06	3.87%	12/12/2017

Class system | Component life cycle



Class system | Component template methods

```
Ext.define("My.app.FilterField", {
    extend      : "Ext.form.TextField",

    initComponents : function() {
        ...
    },

    onRender      : function() {
        ...
    },

    onDestroy     : function() {
        ...
    }
});
```



Component Architecture

Choices...

Component Architecture | Choosing a base class

UI Class

```
<div>  
  foo  
</div>
```

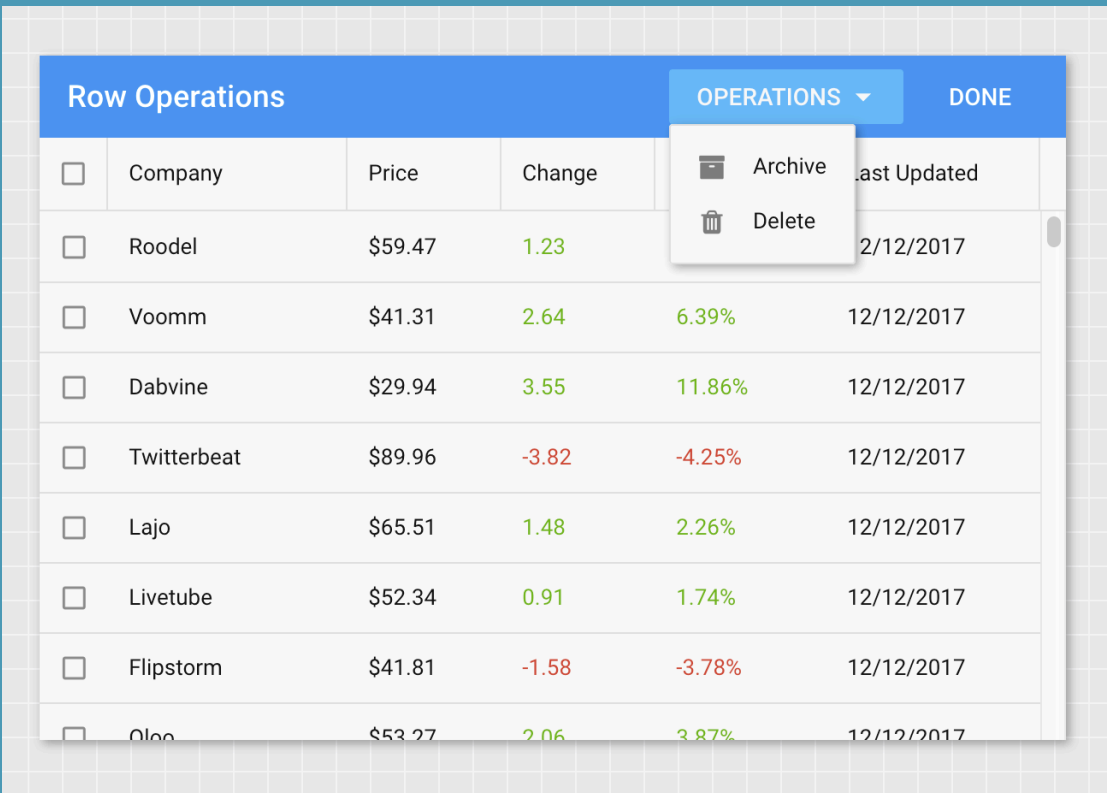
or...

Pure JS class?



Component Architecture | Choosing a UI base class

Extending built-in cmp



A screenshot of a table component. The table has a header row with columns: Company, Price, Change, and Last Updated. Below the header, there are several rows of data. A context menu is open over the first row, showing options: Archive and Delete. The table is styled with a blue header and a light blue background.

Row Operations	Company	Price	Change	Last Updated
<input type="checkbox"/>	Roodel	\$59.47	1.23	12/12/2017
<input type="checkbox"/>	Voomm	\$41.31	2.64	12/12/2017
<input type="checkbox"/>	Dabvine	\$29.94	3.55	12/12/2017
<input type="checkbox"/>	Twitterbeat	\$89.96	-3.82	12/12/2017
<input type="checkbox"/>	Lajo	\$65.51	1.48	12/12/2017
<input type="checkbox"/>	Livetube	\$52.34	0.91	12/12/2017
<input type="checkbox"/>	Flipstorm	\$41.81	-1.58	12/12/2017
<input type="checkbox"/>	Oloo	\$53.27	2.06	12/12/2017

or...

Custom UI cmp?



Non-UI classes could extend **Ext.Base** or **Ext.util.Observable**

```
Ext.define("MyApp.SomeClass", {  
    extend      : "Ext.Base" // Or omit this  
});
```

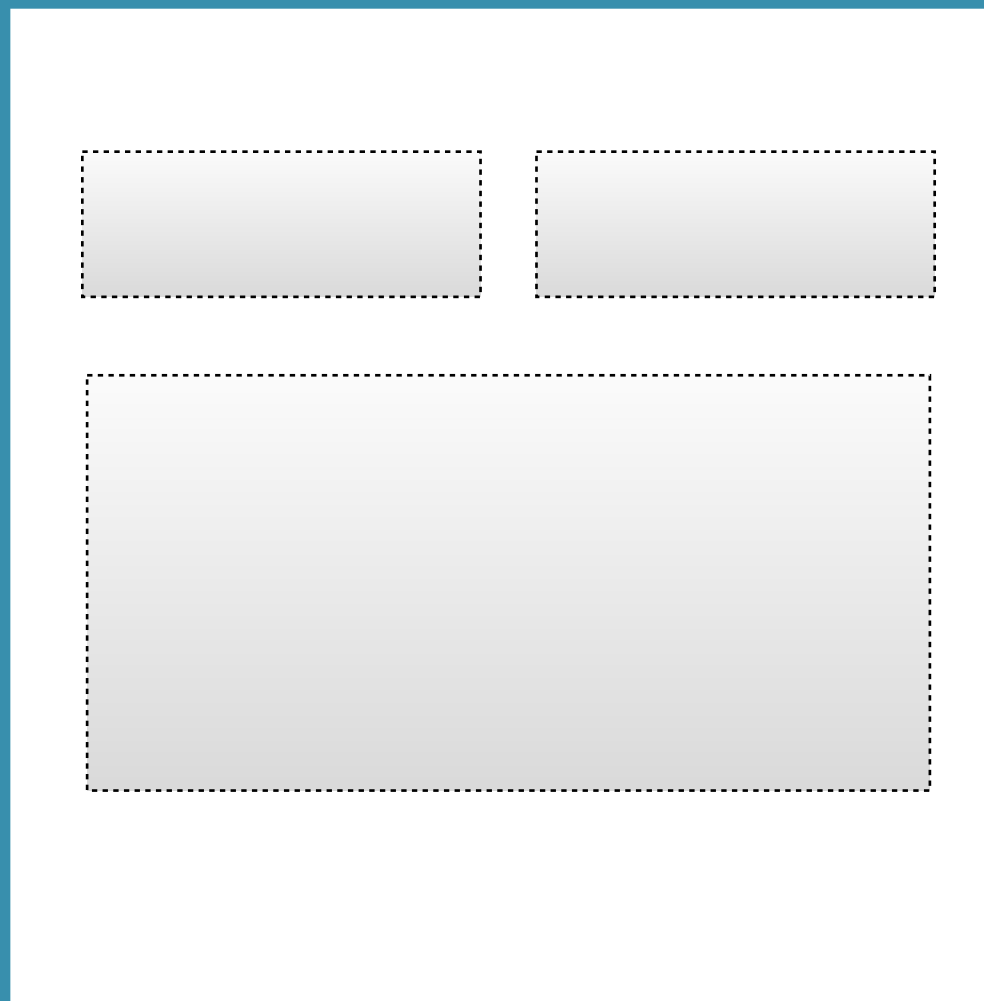
Component Architecture | Choosing a base class

For custom UI classes, you have more choices:

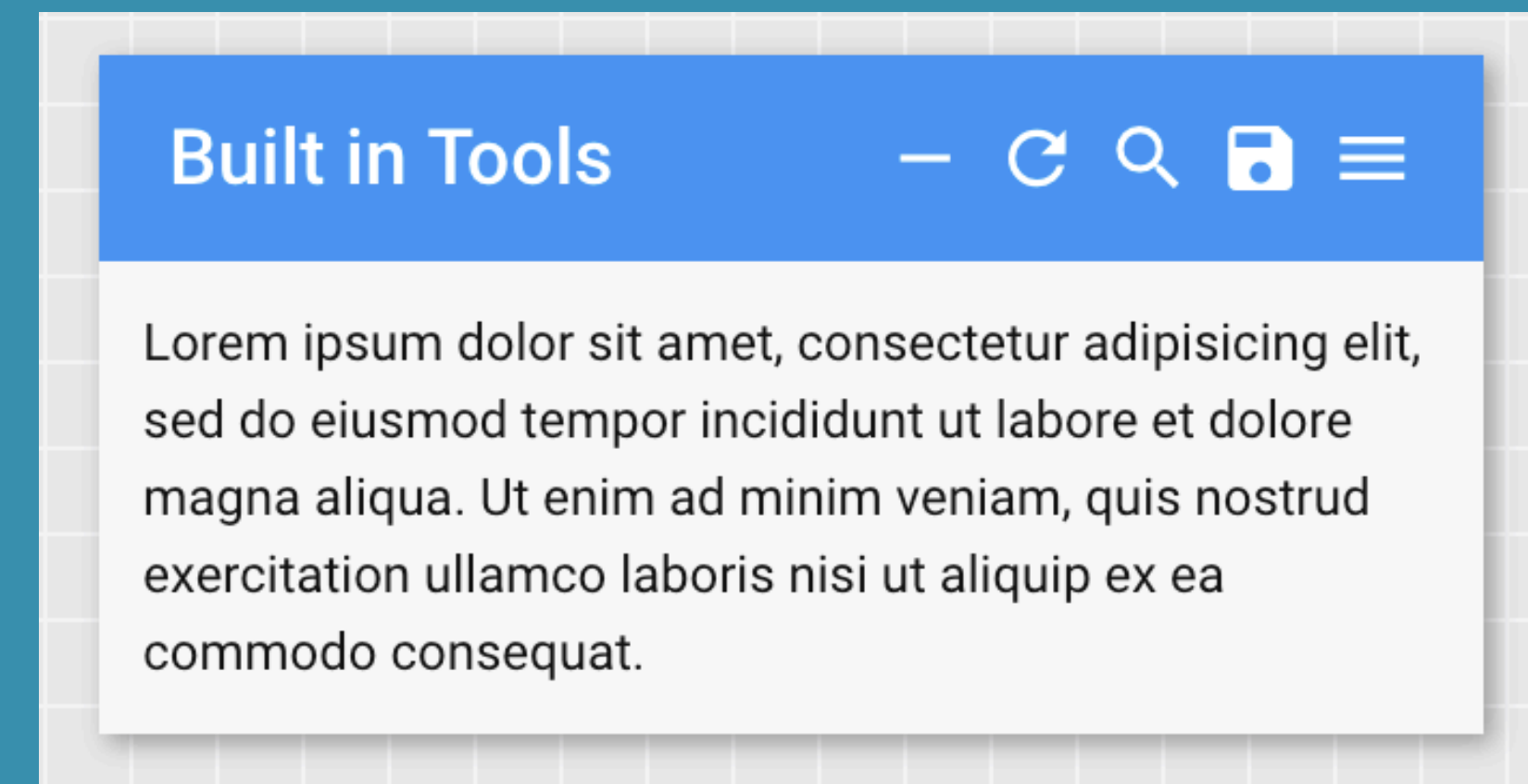
Ext.Component

```
<div>  
  Foo  
</div>
```

Ext.Container



Ext.Panel



Use the right Ext JS abstraction level

Ext.Panel

Title, toolbars, buttons

Ext.Container

Layout, contain other components

Ext.Component

Sizing, life cycle

Ext JS “DOM Core”

JS API, Ext.Element, Ext.get



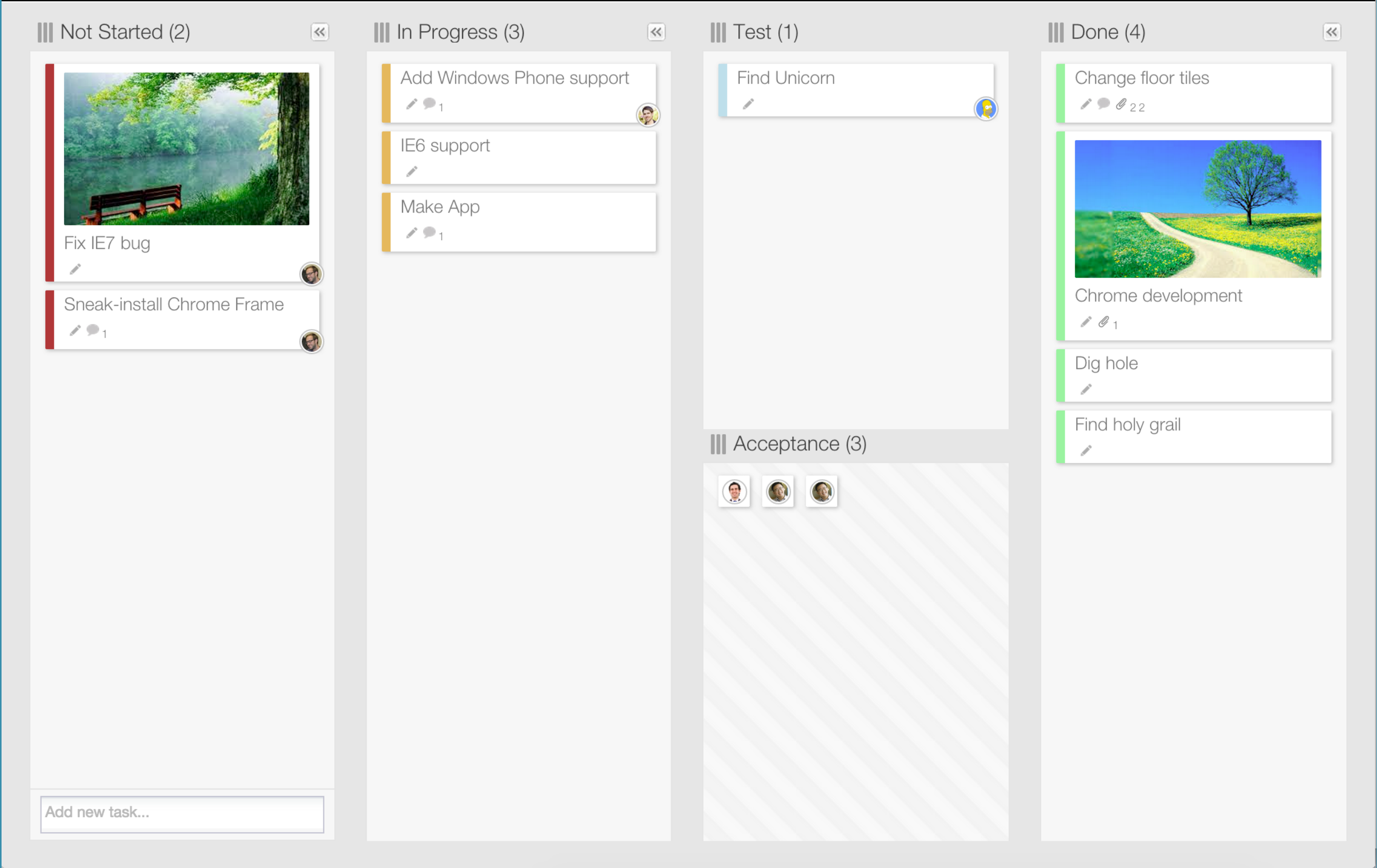
Native DOM

HtmlElement, doc.getElementById

Component Architecture | Break down big components into pieces

- If you're building a complex UI component:
- Break it down into smaller sub-components
- Combine pieces to form a powerful component

Let's look under the hood of the Task Board component



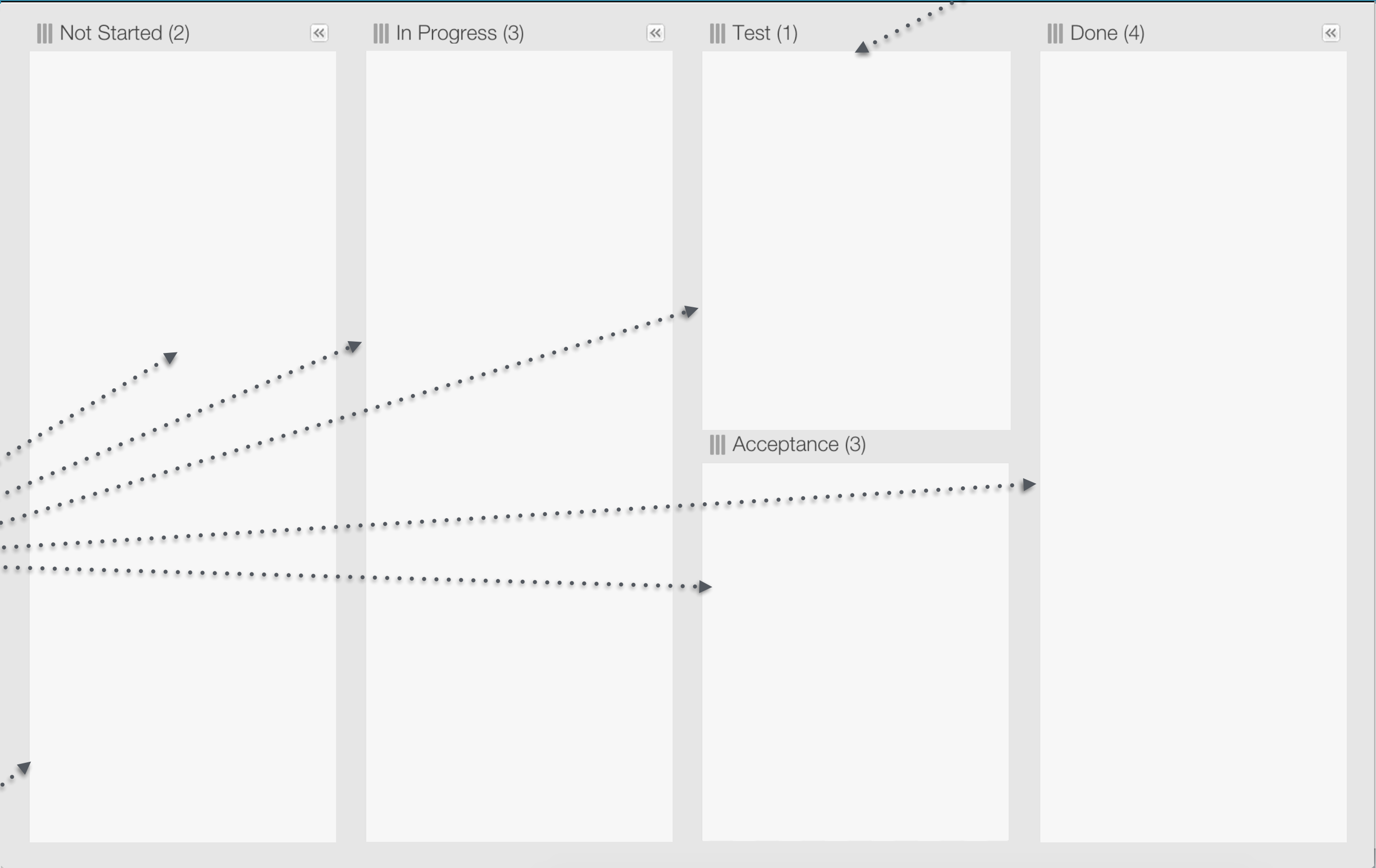
Component Architecture | Task board

Ext.Panel[layout=hbox]

Ext.Container[layout=vbox]

Ext.DataView

Ext.Panel[layout=fit]





Ext Gantt

Mega component for Ext JS

What is a Gantt chart anyway?

Gantt chart

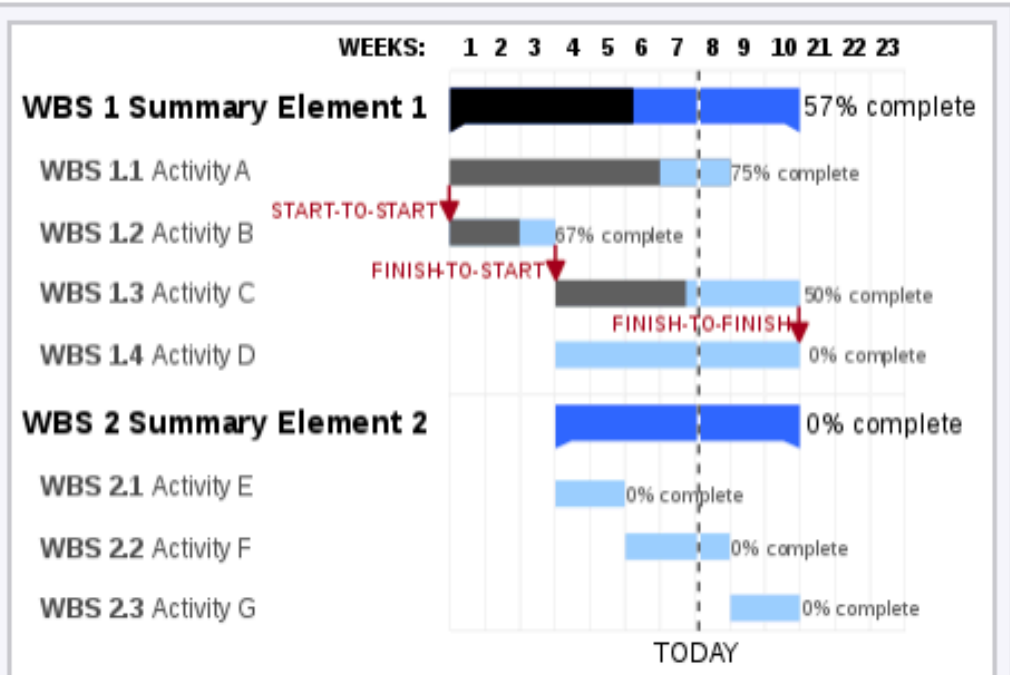
From Wikipedia, the free encyclopedia

"Gantt" redirects here. For other uses, see [Gantt \(disambiguation\)](#).

A **Gantt chart** is a type of [bar chart](#) that illustrates a [project schedule](#), named after its inventor, [Henry Gantt](#) (1861–1919), who designed such a chart around the years 1910–1915.^{[1][2]} Modern Gantt charts also show the [dependency](#) relationships between activities and current schedule status.

Contents [\[hide\]](#)

- 1 [Definition](#)
- 2 [Historical development](#)
- 3 [Example](#)
- 4 [Progress Gantt charts](#)
- 5 [Linked Gantt charts](#)
- 6 [See also](#)
- 7 [Citations](#)
- 8 [References](#)
- 9 [Further reading](#)
- 10 [External links](#)

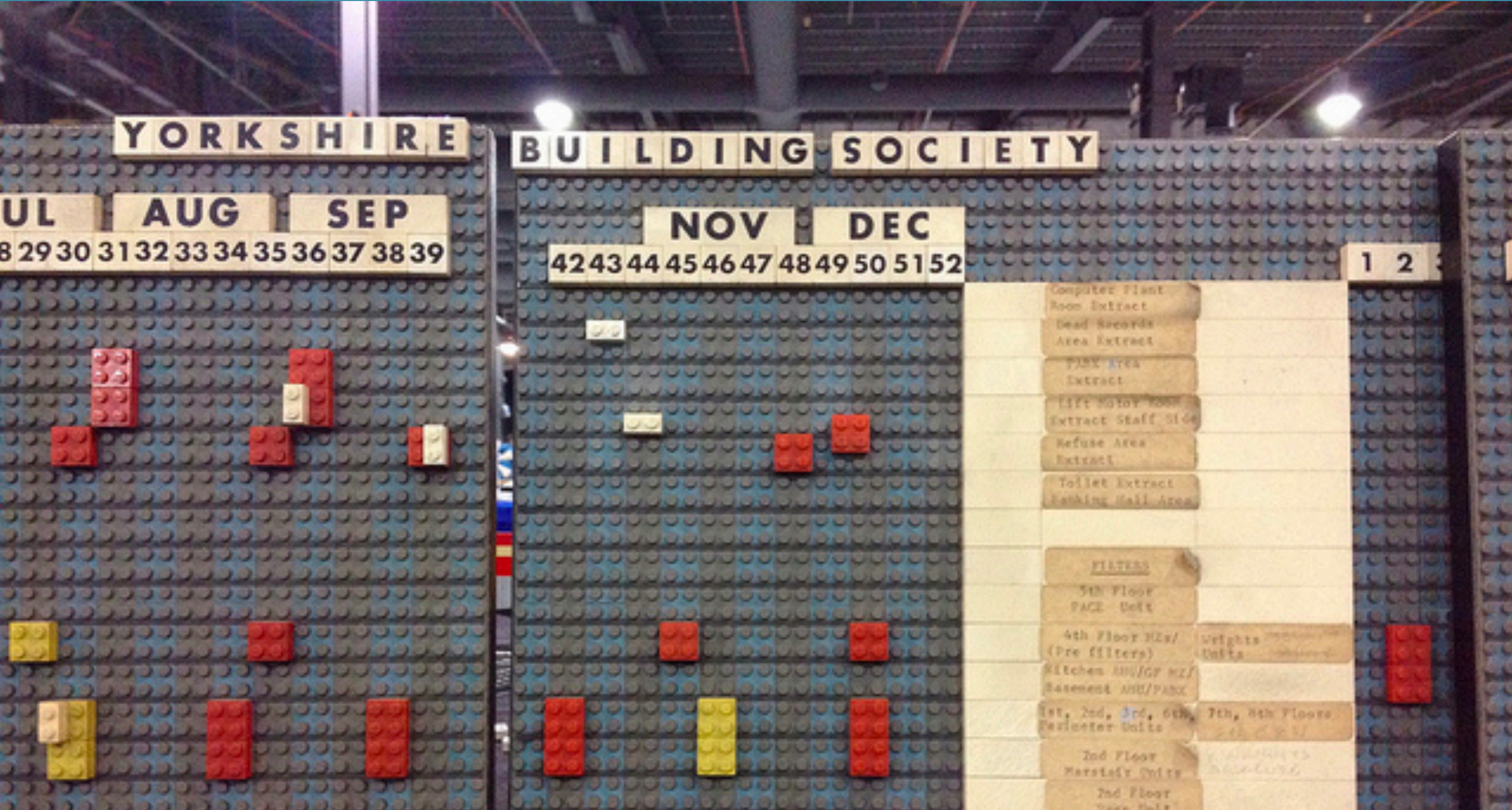


A Gantt chart showing three kinds of schedule dependencies (in red) and percent complete indications.

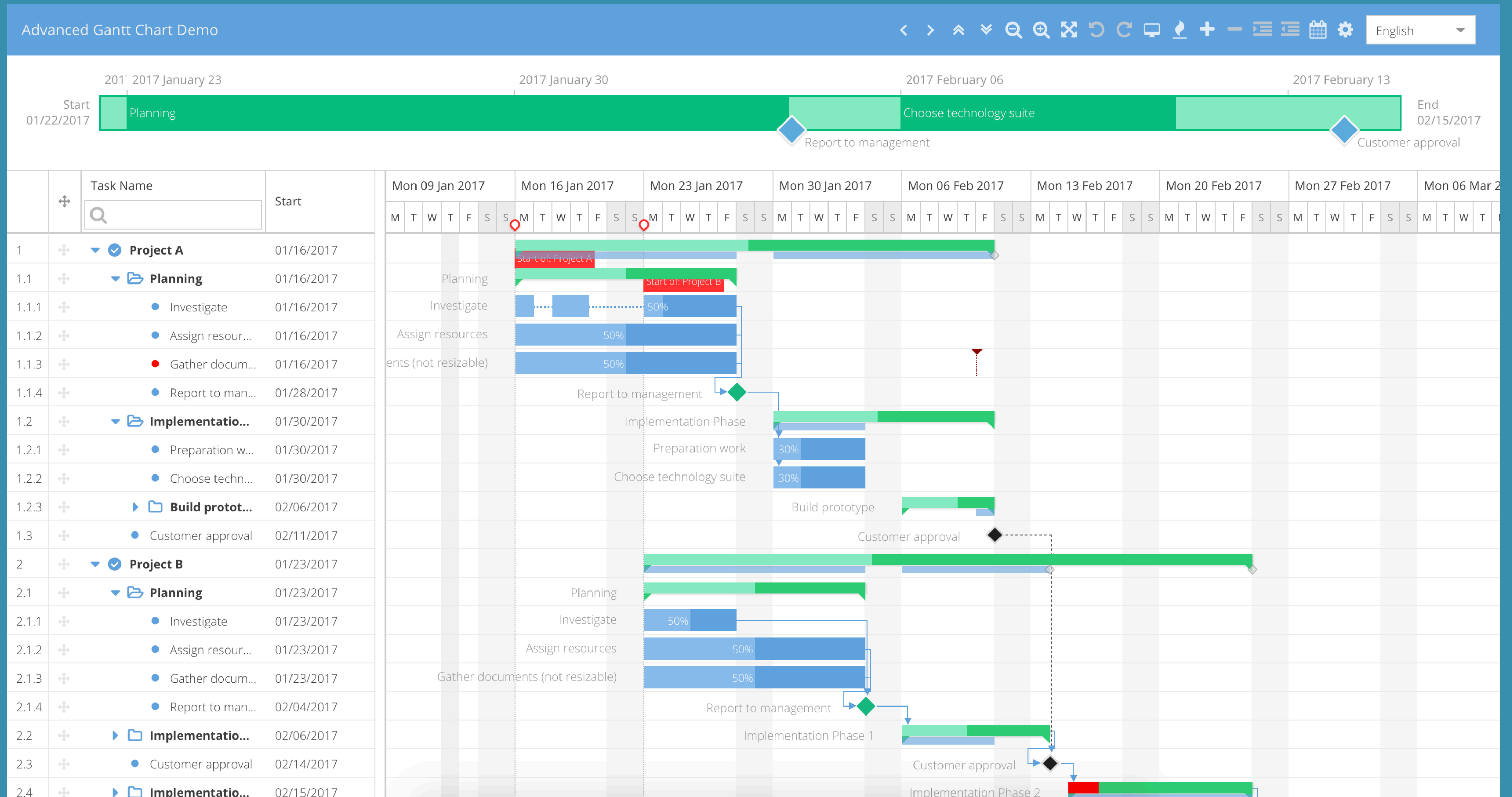
Things a Gantt chart will tell you

- ★ The start date of the project
- ★ What tasks the project consists of
- ★ Who is working on each task
- ★ When tasks start and finish
- ★ How long each task will take
- ★ How tasks group together, overlap and depend on each other
- ★ The finish date of the project

Ext Gantt | Overview



Ext Gantt | Overview



- Ext Gantt is most likely the largest Ext JS component
- 400+ JS files
- gnt-all-debug.js is 1.9MB
- Covered by 600+ test files

BUT: using it is as easy as using any other Ext JS component

```
Ext.create('Ext.Viewport', {
    layout : 'fit',
    items : [
        {
            xtype : 'ganttpanel',
            columns : [
                {
                    xtype : 'namecolumn',
                    width : 250
                }
            ],
            viewPreset : 'weekAndDayLetter',
            taskStore : {
                type : 'gantt_taskstore',
                proxy : {
                    type : 'ajax',
                    url : 'data/tasks.json'
                }
            }
        }
    ]
});
```



Ext Gantt Architecture

Breaking it down

Subclasses the Ext.tree.Panel

Core Team Projects

Export to ...

Task	Duration	Assigned To	Done	Edit
▼ Project: Shopping	13h 15m	Tommy Maintz	<input type="checkbox"/>	
▶ Housewares	1h 15m	Tommy Maintz	<input type="checkbox"/>	
▼ Remodeling	12 hours	Tommy Maintz	<input type="checkbox"/>	
▶ Paint bedroom	2h 45m	Tommy Maintz	<input type="checkbox"/>	
Retile kitchen	6h 30m	Tommy Maintz	<input type="checkbox"/>	
Decorate living room	2h 45m	Tommy Maintz	<input checked="" type="checkbox"/>	
Fix lights	45 mins	Tommy Maintz	<input checked="" type="checkbox"/>	
Reattach screen door	2 hours	Tommy Maintz	<input type="checkbox"/>	
▶ Project: Testing	2 hours	Core Team	<input type="checkbox"/>	

- The Ext JS TreePanel uses one **Store** for its rows
- Ext Gantt uses six different stores
 - Tasks
 - Dependencies
 - Assignments
 - Resources
 - Calendars
 - Lines / Zones

Ext Gantt | Architecture

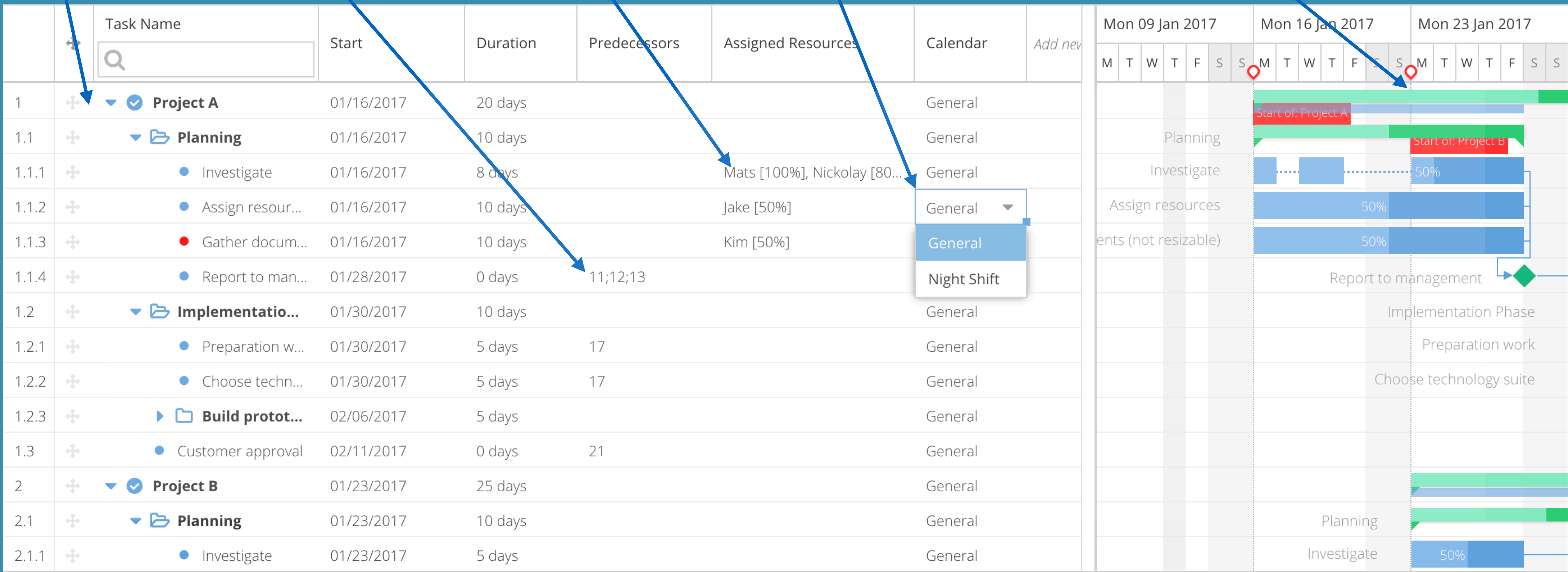
Tasks

Dependencies

Resources + Assignments

Calendars

Time lines



- Built with Sencha Cmd
- Tested by Siesta, TeamCity, SauceLabs, BrowserStack
- Composed by plenty of plugins / mixins
- 5 themes supported
- Cmd Package available

Ext Gantt utilises almost every part of Ext JS:

- GridPanel / TreePanel
- Window
- Tooltips
- Forms
- Stores
- Ajax
- TabPanel



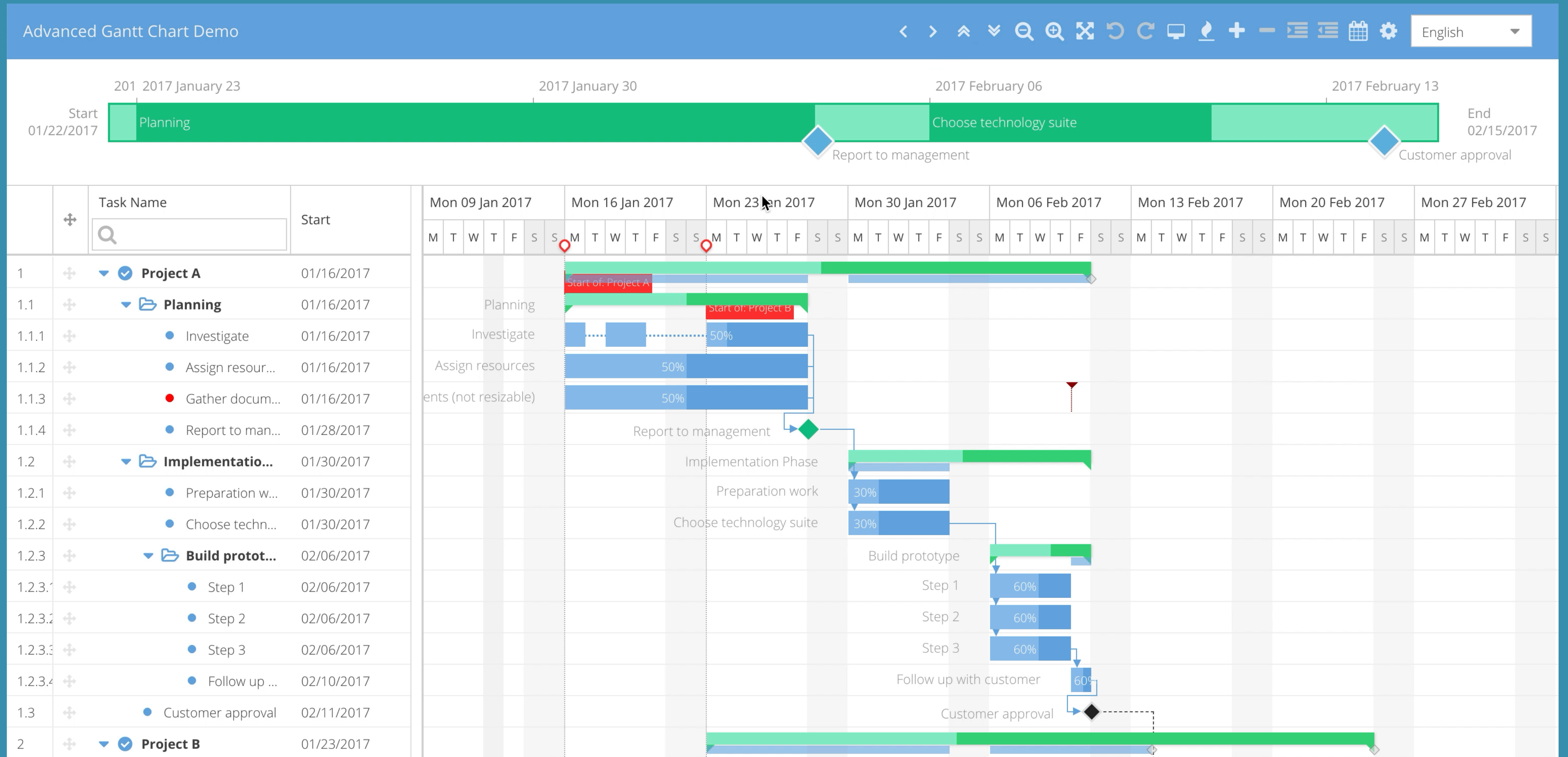
Scheduling Engine

Keeping tasks aligned

- A Gantt chart is much more than a project visualisation
- Tasks are placed on the timeline based on many factors:
 - predecessors
 - constraints
 - task calendar
 - scheduling mode
- Changes cascade onto any dependent tasks

- The **TaskStore** in Ext Gantt is not just for containing data
- It also serves as the scheduling engine to keep tasks aligned
- Changing one task may cause an entire project to be rescheduled
- The TaskStore subclasses Ext.data.TreeStore
- Logic both on Store and Model level => **very** easily tested

Ext Gantt | Scheduling Engine





Adding Ext Gantt to an Ext JS application

Just another reusable component

Ext Gantt | Adding to an Ext JS application

- Using a mega component is still as easy as using a TextField
- If you are familiar with Ext JS **Grid** / **Store** / **Model** it's very easy
- Use the 'ganttpanel' xtype or **Gnt.panel.Gantt** constructor
- Either build using Sencha Cmd or include gnt-all.js

Ext Gantt | Adding to an Ext JS application

```
<!DOCTYPE HTML>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8">

  <link href="../../../extjs-6.5.3/build/classic/theme-triton/resources/theme-triton-all.css" rel="stylesheet" type="text/css"/>
  <link href="../../../resources/css/sch-gantt-triton-all.css?ver=%VERSION%" rel="stylesheet" type="text/css" />

  <script src="../../../extjs-6.5.3/build/ext-all-debug.js" type="text/javascript"></script>
  <script src="../../../extjs-6.5.3/build/classic/theme-triton/theme-triton.js" type="text/javascript"></script>

  <script src="../../../gnt-all-debug.js?ver=%VERSION%" type="text/javascript"></script>

  <script src="app.js" type="text/javascript"></script>

  <title>Basic Gantt demo</title>
</head>
<body>
</body>
</html>
```

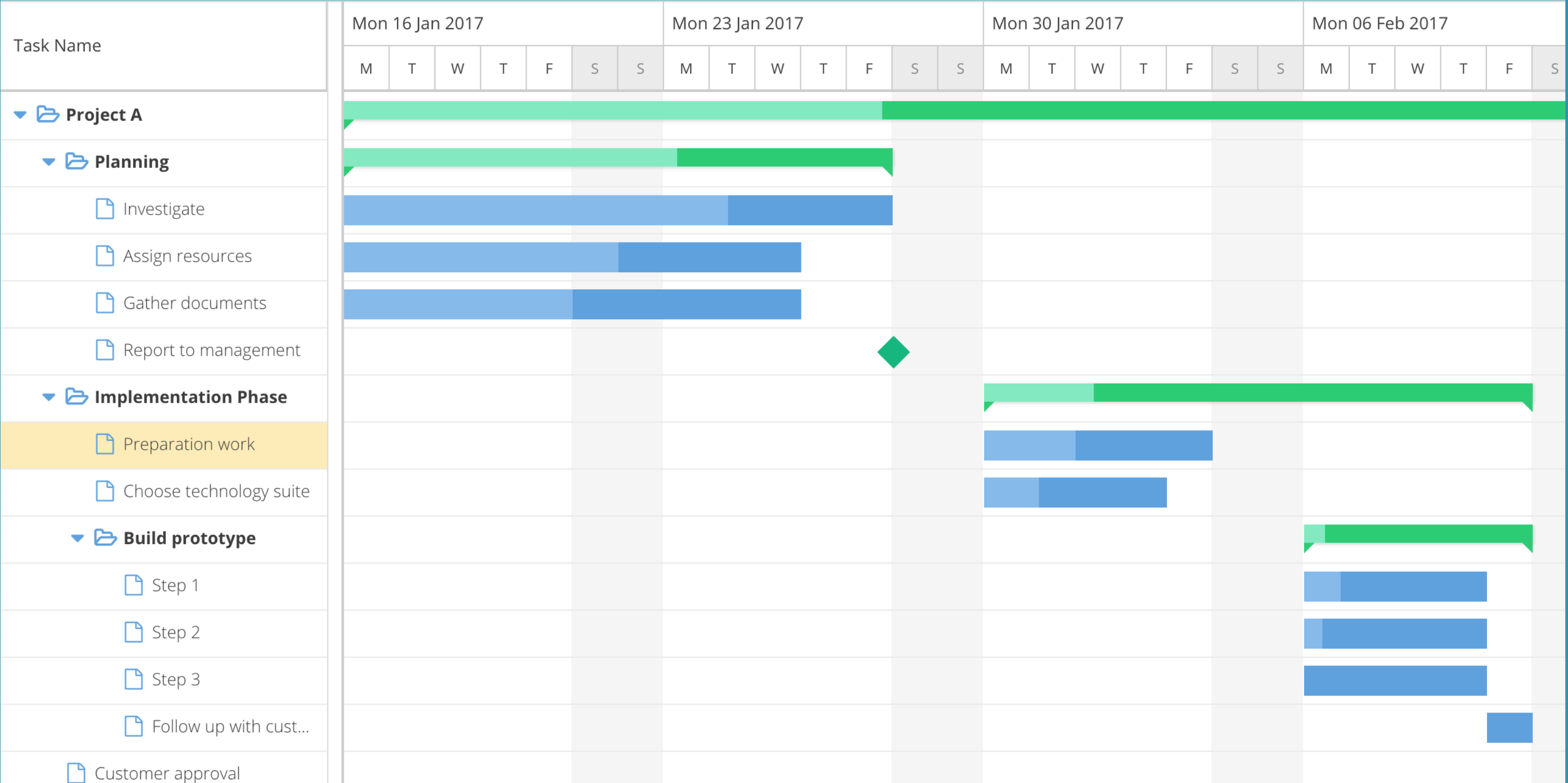
Ext Gantt | Adding to an Ext JS application

```
Ext.create('Ext.Viewport', {
    layout : 'fit',

    items : [
        {
            xtype      : 'ganttpanel',
            rowHeight  : 35,
            columns : [
                {
                    xtype : 'namecolumn',
                    width : 250
                }
            ],

            viewPreset : 'weekAndDayLetter',

            taskStore : {
                type : 'gantt_taskstore',
                proxy : {
                    type : 'ajax',
                    url : 'data/tasks.json'
                }
            }
        }
    ]
});
```



Let's add some dependencies

Ext Gantt | Adding to an Ext JS application

```
Ext.create('Ext.Viewport', {
    layout : 'fit',


    items : [
        {
            xtype      : 'ganttpanel',
            rowHeight : 35,

            columns    : [
                {
                    xtype : 'namecolumn',
                    width  : 250
                }
            ],

            viewPreset : 'weekAndDayLetter',

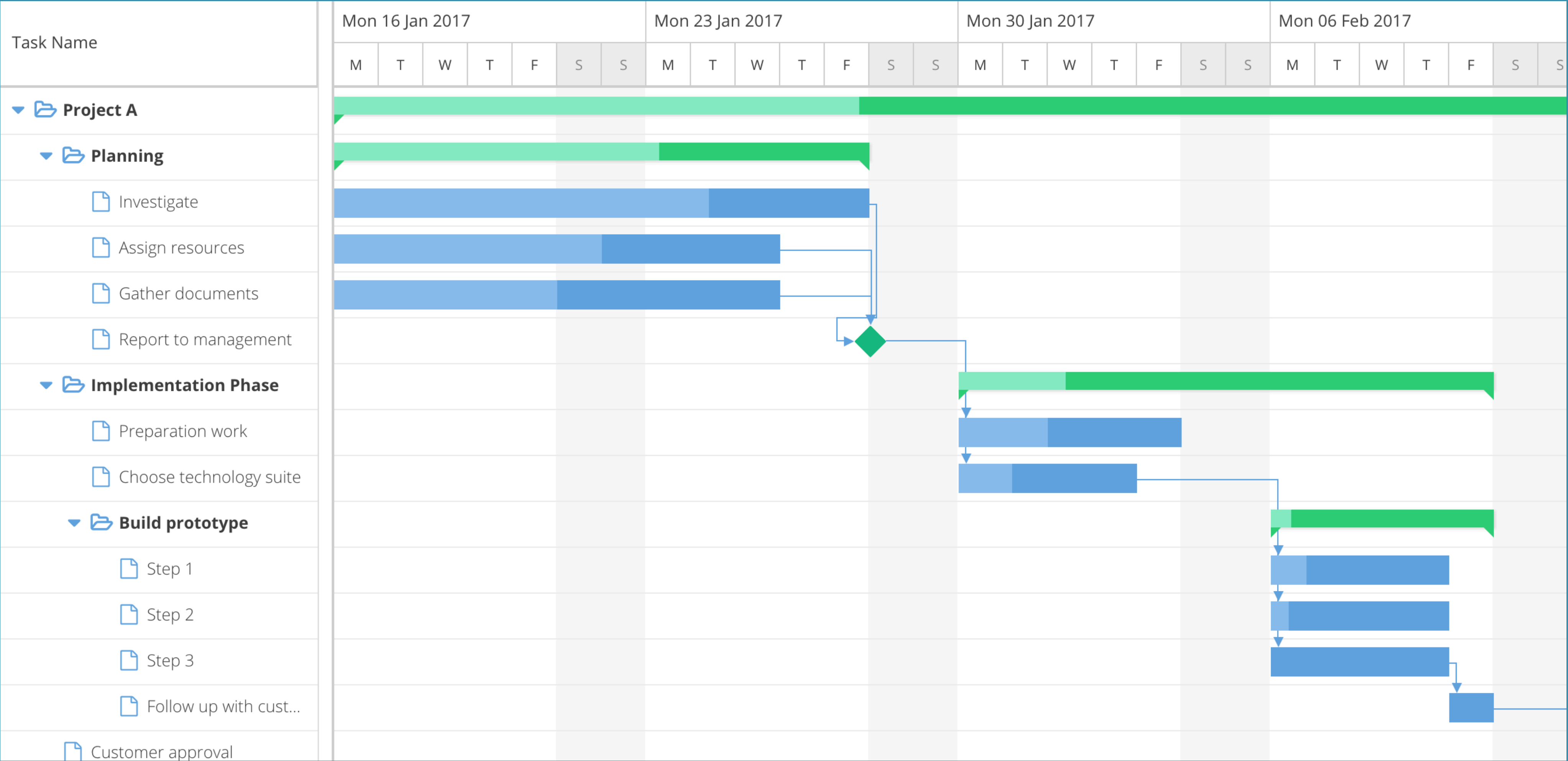
            taskStore : {
                type : 'gantt_taskstore',
                proxy : {
                    type : 'ajax',
                    url  : 'data/tasks.json'
                }
            },

            dependencyStore : {
                type           : 'gantt_dependencystore',
                autoLoad       : true,
                proxy          : {
                    type : 'ajax',
                    url  : 'data/dependencies.json'
                }
            }
        }
    ]
});
```



```
[
    {
        "Id"   : 1,
        "From" : 11,
        "To"   : 17
    },
    {
        "Id"   : 2,
        "From" : 12,
        "To"   : 17
    }
]
```

Ext Gantt | Adding to an Ext JS application



DEMO: Combination with other Components

DEMO: Various use cases inside Ext components

DEMO: Inception tooltips



Tips & Tricks

Useful bits of info

- Stay on top of the public Ext JS API as much as possible
- Upgrade pains to be expected otherwise
- If you must override a private Ext JS method, **document** it
- At Bryntum, we detect private overrides through **tests**
- Patches constrained to specific versions of Ext JS

Writing testable code

Models & Stores are pure JS, not tied to DOM

Very easy to test. Very fast.

Views & controllers are much harder

Rounding up

- Break your application into reusable Components
- Test your big components individually
- Even a huge component can be easily reused in an Ext JS app

- sencha.com/extjs
- bryntum.com/products/
- therootcause.io

Questions?

Twitter: @Bryntum