Building desktop apps with JS

by

Chris Rickard

@chrisrickard

http://inoutput.io

So why the hell would you do it?

(build desktop apps with webtech)

So compared to a "classic" desktop app...

Objective-C

Great frameworks...

YEOMAN

jQuery
write less, do more.

Leaflet

CoffeeScript

ember

Bootstrap

Pjs

d3

BACKBONE.JS

THREE.JS

{less}

Haml

handlebars

Sass.

RIVETS.JS

ANGULARJS
by Google

Derby

Ext JS

Cappuccino

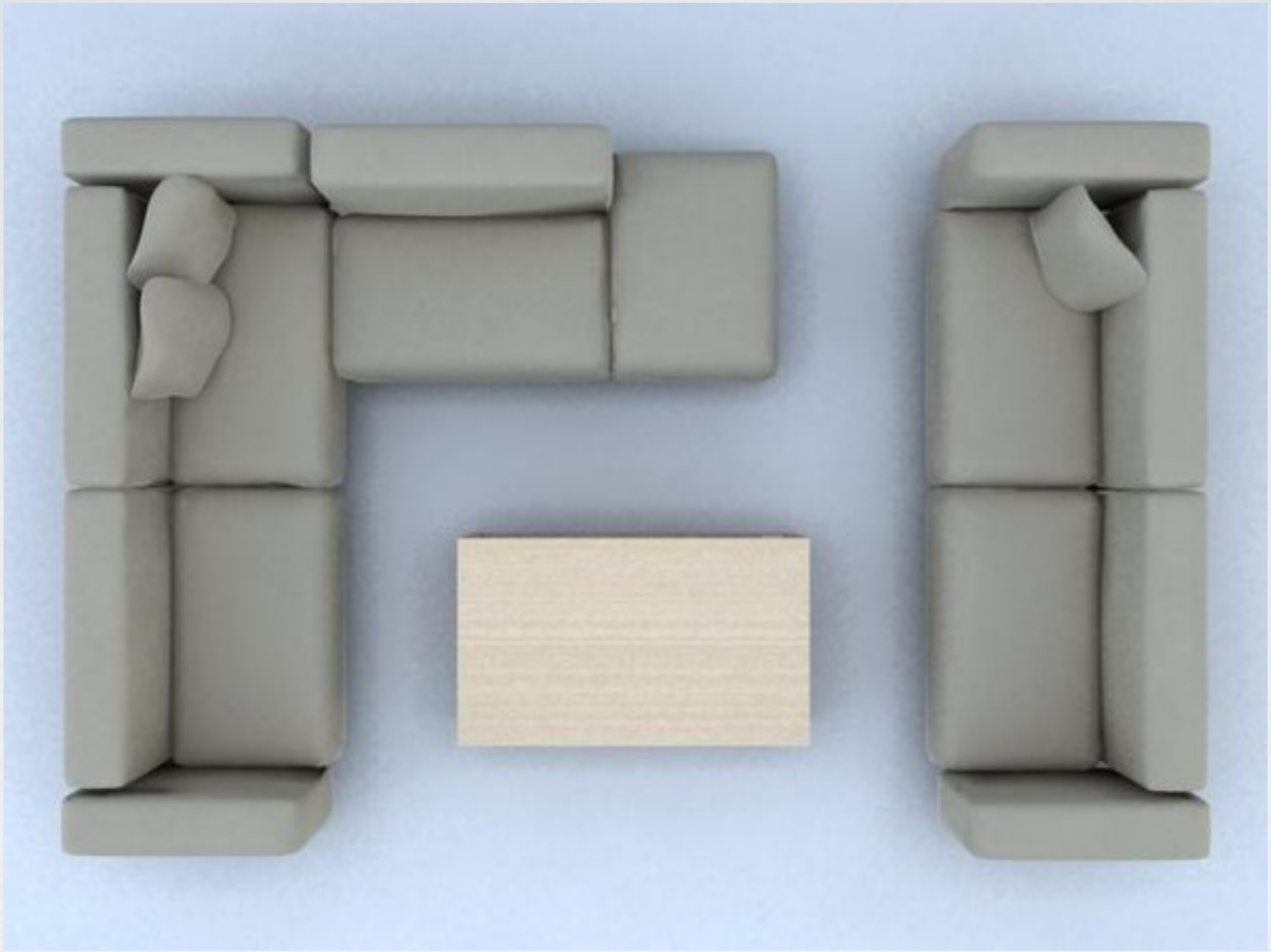
dōjō

Knockout.

SPROUTCORE

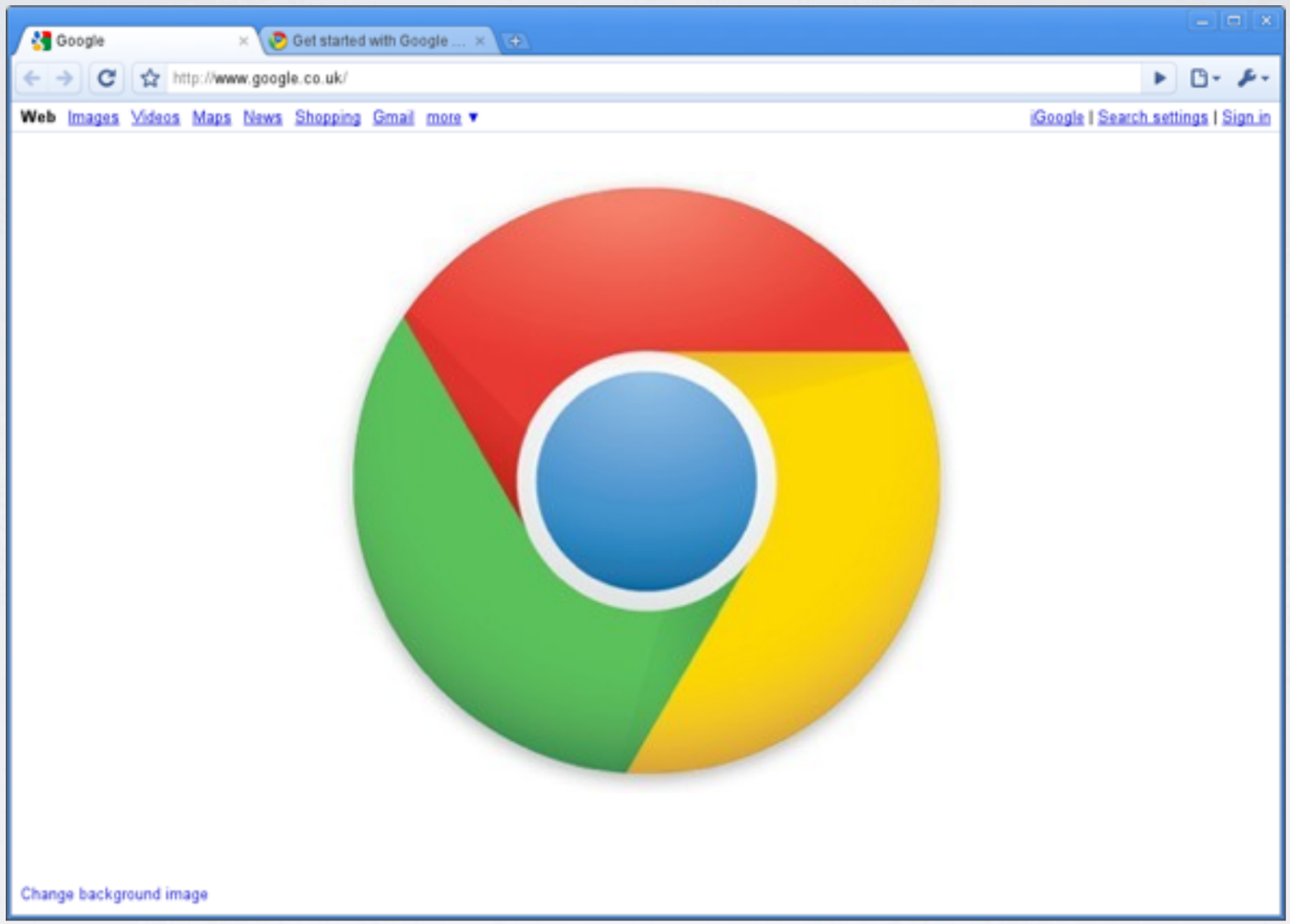
Meteor JS

Modularity...



Better the devil you know...

... and compared to a Webapp?



Power

- Access to underlying OS
- File menu, system tray etc..
- No security restrictions
- Local file access
- No prompts for camera, audio, location etc
- No cross domain restrictions

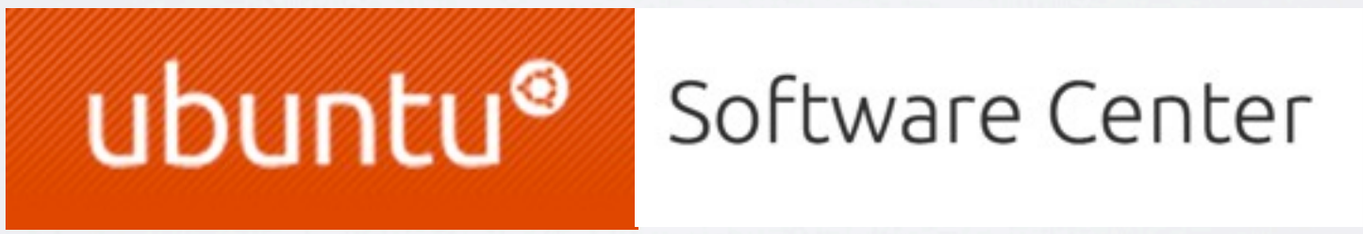
Compatibility

Just this

and not all of these suckers:

Distribution

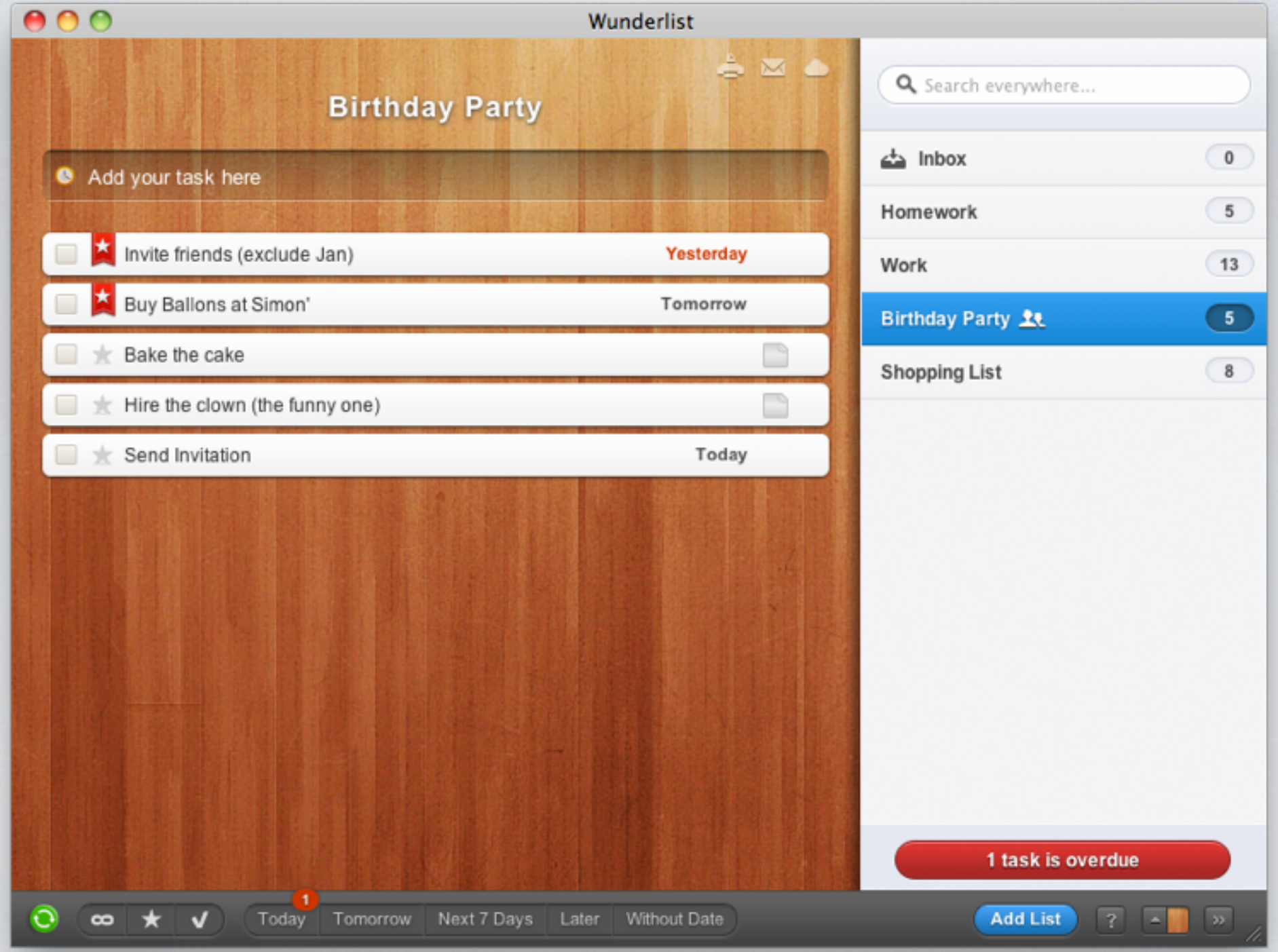
Windows 8 Store



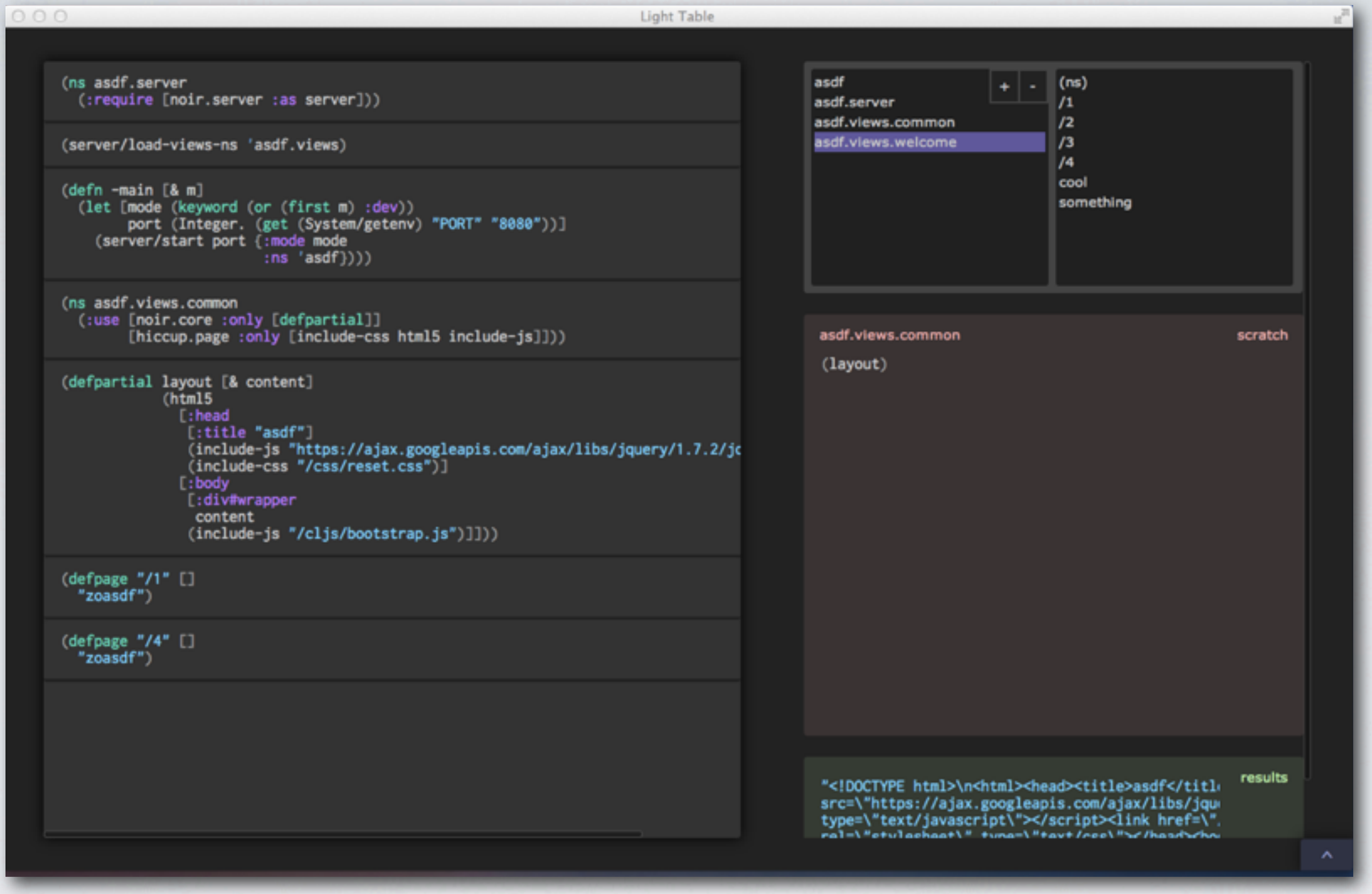
blah blah blah....

So who are building their apps in JS?

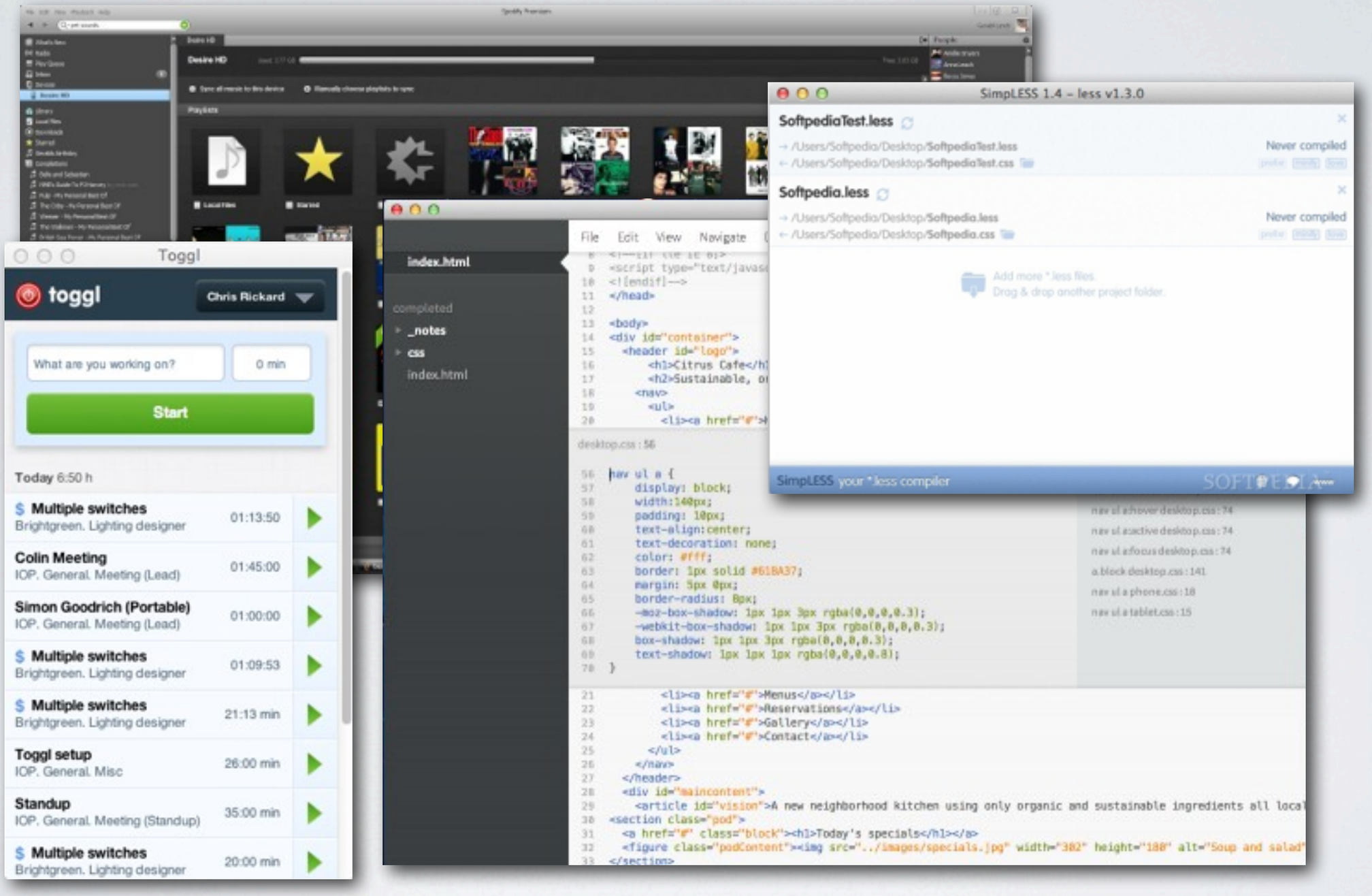
Wunderlist


Wunderlist
Birthday Party
Add your task here
Invite friends (exclude Jan)
Yesterday
Buy Ballons at Simon'
Tomorrow
Bake the cake
Hire the clown (the funny one)
Send Invitation
Today
Search everywhere...
Inbox
0
Homework
5
Work
13
Birthday Party
5
Shopping List
8
1 task is overdue
1
Today
Tomorrow
Next 7 Days
Later
Without Date
Add List

LightTable IDE



...and a bunch of others



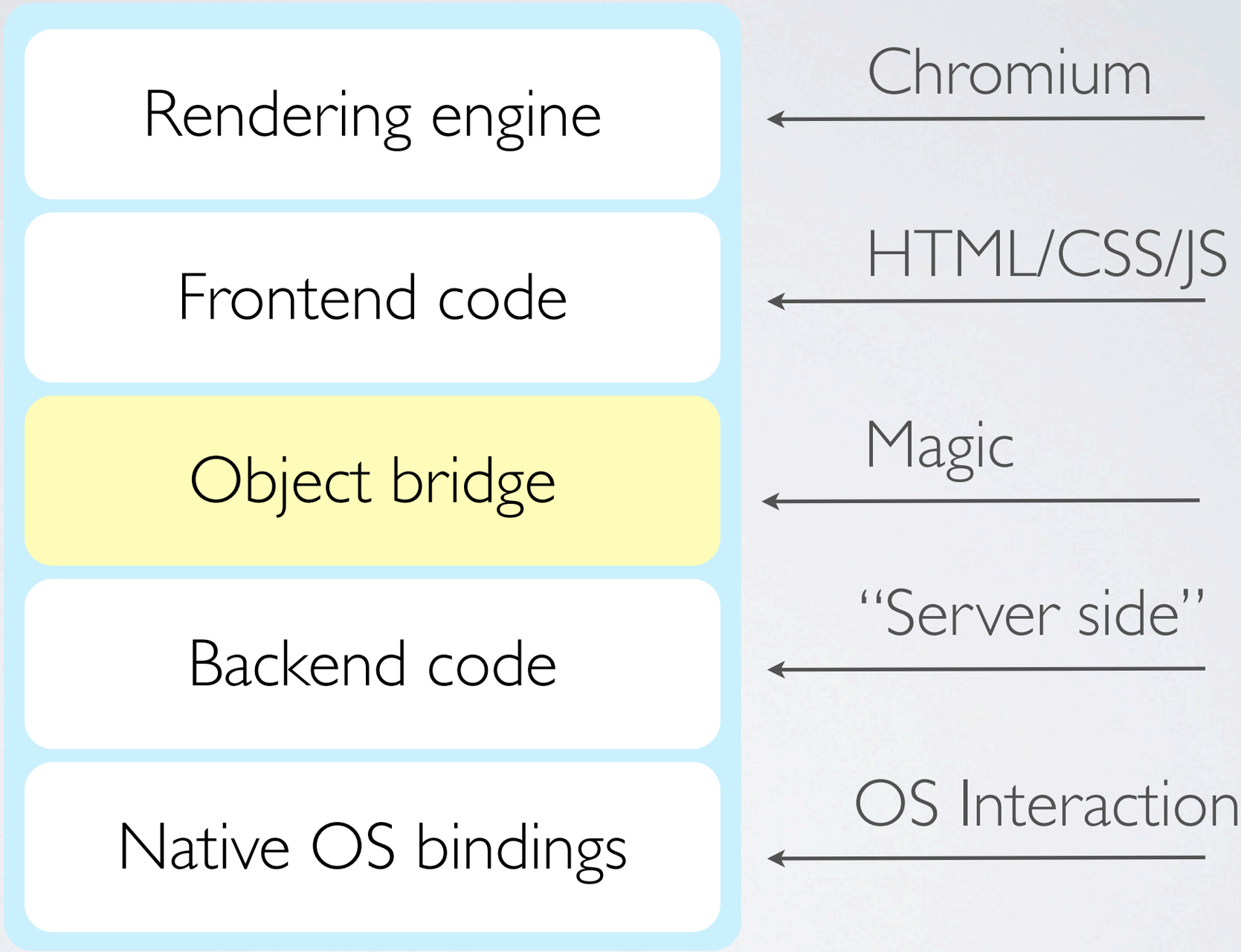
The current landscape

- AppJS
- node-webkit
- TideSDK

- Sencha Desktop Packager (osx & win)
- Adobe AIR (osx & win... and it's flash)

or Chromium Embedded Framework (CEF)

So how does it all work?



- Was previously "Titanium Desktop"
- Supports Ruby, Python & PHP
- Creates a bridge between chromium and libtide - marshalling, object mapping etc.
- After Titanium ditched it - picked up by OS
- Possibly needs a little bit more time...

http://www.tidesdk.org

- NodeJS
- Built on CEF (Chromium embedded framework)
- Bridges Chromium & node via IPC
- Does talk "across the wire" - but hooks into the custom appjs:// scheme to handle requests

http://appjs.org

nodewebkit

- NodeJS
- Rad. Merged Node & Chromiums Event-loops
- Direct access, same thread
- Sync's with newest Chromium build every few days
- Backed by Intel

https://github.com/rogerwang/node-webkit

nodewebkit

Let's take a look at a demo...

And the code?

The beginning: package.json

```
{
  "name": "Example",
  "main": "index.html",
  "window": {
    "name": "example",
    "title": "Example",
    "position": "center",
    "width": 700,
    "height": 500,
    "toolbar": false,
    "show": false
  }
}
```

And the code?

Adding a menubar

```
1  var gui = require('nw.gui');
2
3  var main_menu = new gui.Menu({ type: 'menubar' });
4  var submenu = new gui.Menu();
5
6  submenu.append(new gui.MenuItem({ type: 'checkbox', label: 'You' }));
7  submenu.append(new gui.MenuItem({ type: 'checkbox', label: 'Can' }));
8  submenu.append(new gui.MenuItem({ type: 'checkbox', label: 'Tick' }));
9  submenu.append(new gui.MenuItem({ type: 'checkbox', label: 'These items' }));
10
11 main_menu.append(new gui.MenuItem({ label: 'Menu', submenu: submenu }));
12 main_menu.append(new gui.MenuItem({ icon: 'img/cut.png', label: 'Cut' }));
13 main_menu.append(new gui.MenuItem({ icon: 'img/apple.png', label: 'Play' }));
14 main_menu.append(new gui.MenuItem({ icon: 'img/tick.png', label: 'Tick' }));
15
16 gui.Window.get().menu = main_menu;
```

And the code?

External process interaction

```
1  var child_process = require('child_process');
2  var proc = process.exec('ping -c 5 reddit.com');
3
4  proc.stdout.on('data', function (data) {
5    console.log(data);
6  });
```

Packaging? Easy

- Zip app directory, and rename app.nw
- OSX - Copy node-webkit.app & place app.nw inside
- Windows - copy /b nw.exe+app.nw app.exe
- Linux - cat /usr/bin/nw app.nw > app && chmod +x app

but... apps are 24MB (compressed) for a hello world :(

And thats pretty much it...

1. If your building a desktop app: consider using JS
2. A bunch of neat frameworks already out there
3. But personally I think Webapps will eventually win

Thanks... Questions?!

@chrisrickard