



BALTIMORE

— DRUPALCON 2017 —

*Supercharge your next web app
with Electron!*



GET /user/techninja



FOUR
KITCHENS

James Todd

- Drupal Dev for 10+ years from Nor Cal
- Full time developer at Four Kitchens
- Maker, promoter of STEAM for kids
- Co-Author for Arduino Book series
- Maintainer of 2 OSS Electron apps for robotic control (*check out a demo our booth!*)





**Hold questions until the end
Thank you :)**

Quick Term Reference

- **JavaScript:** Programming language for web browsers
- **Node.js:** JavaScript for servers
- **NPM:** The node package manager
- **API:** Interface to let programs to something
- **ReST/ful:** Standard for APIs over the web

Quick Term Reference

- **DOM:** Object model for web page elements
- **React:** Web framework for state & UI management
- **Express:** node.js “web server” and ReST router
- **Headless:** Running just a backend without a frontend



What is Electron?

Electron is yours!



=



+



Electron is three things...



=



Chromium
for making
web pages

+



Node.js
for filesystems
and networks

+



Native APIs
for three
systems



Electron's architecture makes it better

Main Process

Renderer Processes



Chrome File Edit View History

The diagram illustrates the Electron architecture. On the left, a box labeled 'Main Process' contains a menu bar with 'Chrome', 'File', 'Edit', 'View', and 'History'. On the right, a larger box labeled 'Renderer Processes' contains three smaller white boxes, each with a black title bar and a single black window control button (minimize, maximize, or close). The background is a dark, stylized image of a city skyline with a prominent church dome and a tall chimney.

Creating with Electron means:

- Single codebase, cross platform native
- Use standard HTML 5 markup
- Use Bootstrap, Angular, React, jQuery and more
- Incredibly powerful debugging w/Chrome dev tools
- Only *one* browser version to target!
- No need to worry about download times for local files



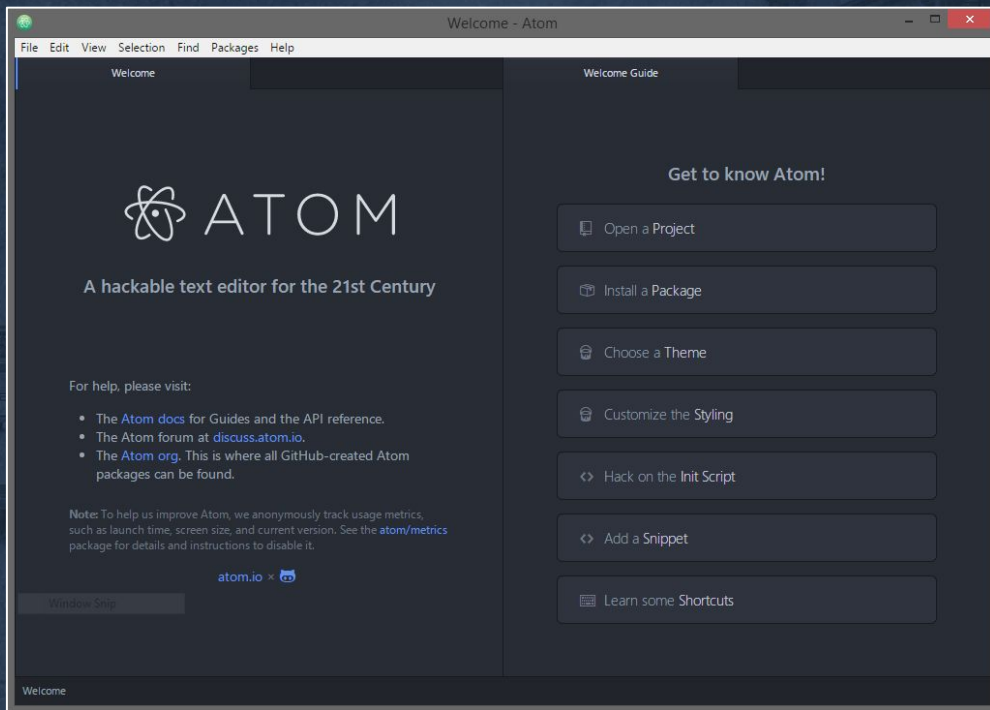
Who uses Electron?



Atom Editor

GitHub

- Infinitely hackable
- > 5k community packages to extend functionality
- > 1mm monthly users
- Dogfooding ensures purity and strong releases

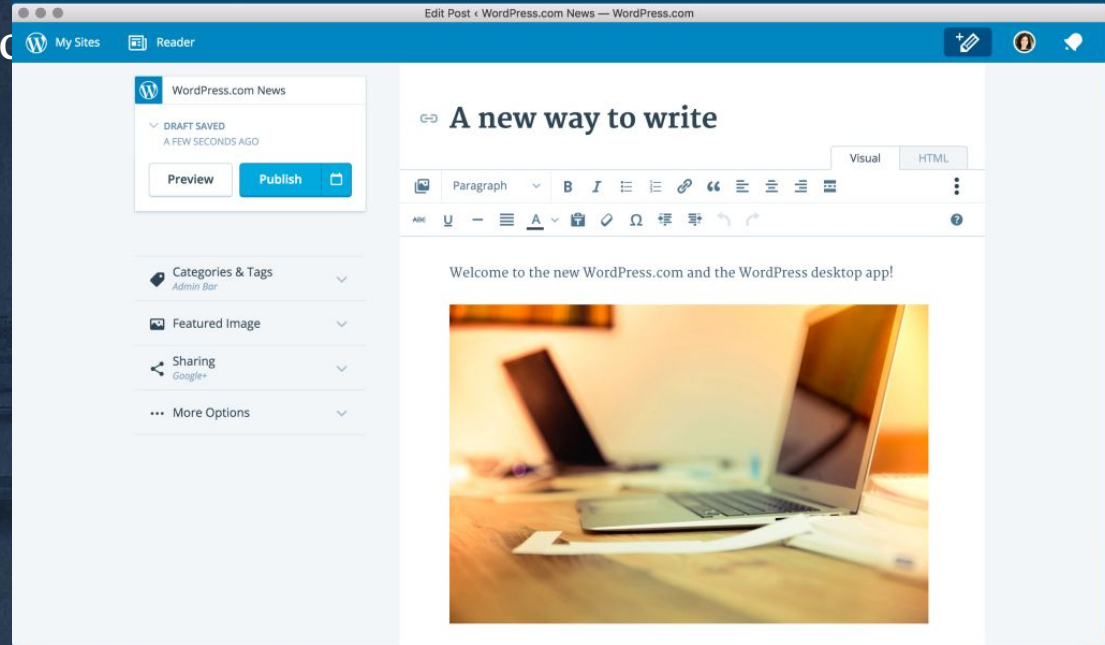




WordPress Desktop

AUTOMATTIC

- Clean, uncluttered interface
- Built by small team in about 2 months
- Direct API access to remotes sites, or offline storage synced.





- MS Open Source
- Different take on code editing
- Live code execution/debugging for node/php & more

The image shows a Visual Studio Code window with the file 'AutomaticArchiveNavigator.lua' open. The file path is 'C:\Users\Public\Documents\AutomaticArchiveNavigator.lua'. The editor displays the following Lua code:

```
1 -----
2 -- AutomaticArchiveNavigator
3 --
4 -- This module produces a talk archive banner, together with an automatically-
5 -- generated list of navigation links to other archives of the talk page in
6 -- question. It implements {{Automatic archive navigator}} and
7 -- {{Talk archive navigation}}.
8 -----
9
10 local yesno = require("Module:Yesno")
11
12 -----
13 -- Helper functions
14 -----
15
16 local function makeWikilink(page, display)
17     if display then
18         return string.format('[[%s|%s]]', page, display)
19     else
20         return string.format('[[%s]]', page)
21     end
22 end
23
24 local function escapePattern(s)
25     -- Escape punctuation in a string so it can be used in a Lua pattern.
26     s = string.gsub(s, '%p', '\000')
27     return s
28 end
29
30 -----
31 -- Navigator class
32 -----
33
34 local Navigator = {}
35 Navigator._index = Navigator
36
37 function Navigator.new(args, cfg, currentTitle)
38     local obj = setmetatable({}, Navigator)
39
40     -- Set inputs
```


electron.atom.io/apps



1Clipboard



Beaker Browser



Brave Browser



Caret



Collectie



Flow



Ghost



GitKraken



Hyper



Insomnia



JIBO



Kap



Kitematic



Marp



Now



Nylas N1



Ramme



Simplenote



Slack



Svgsus



WebTorrent

Quando a Roma...

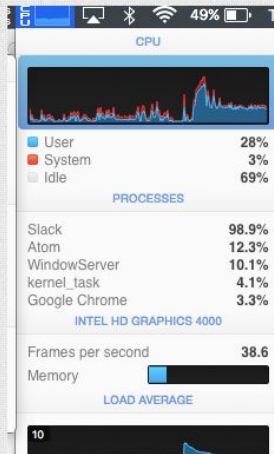


???

29 Oct 2016

Electron is flash for the desktop

What is slack doing?



The process was in the background when this happened. I wasn't even interacting with it - I was in a meeting. I only noticed because my laptop fans were whurring when I got back. Restarting slack seemed to fix it for now.

“People are excited about the Desktop again...”

–Paul Betts

▲ xpaulbettsx 6 days ago [-]

Here's the thing. You know what the alternative to all of these Electron apps coming out is? If your answer is "A native Cocoa/WPF app", you are on another planet, the answer is, "It wouldn't exist at all".

Nobody in the last 5-10 years cared about writing Desktop apps before Electron came along, there's basically zero money in it, and it's massively expensive, both in terms of actual dev time per feature (easily 10x the cost), and also in finding specialist developers who know these dated technologies. And as for Qt, Qt has existed for *over two decades* - if its massive "Beatles walking off the plane" moment hasn't happened by then, sorry, it's not gonna.

But now? People are making all kinds of great new apps, and more often than not, they come out on all three platforms. People are *excited* about the Desktop again - Electron is so good it's single-handedly revitalizing the platform that two of the largest tech companies in the world are behind, yet couldn't do.

That is a Big Deal.

Electron might NOT be for your app if...

...you answer NO to any of the following:

- Can your app run reliably and be useful while offline?
- Can your team meet user needs for application updates and OS specific install support?
- Does extending to desktop meet few additional needs and use cases beyond native notifications and interfaces?

Electron might be RIGHT for your app if...

- Do you need low level functionality not handled by existing APIs?
- Do you need to port a node or other server side app for client side use?
- Do you need extra async processing power beyond webworkers ?

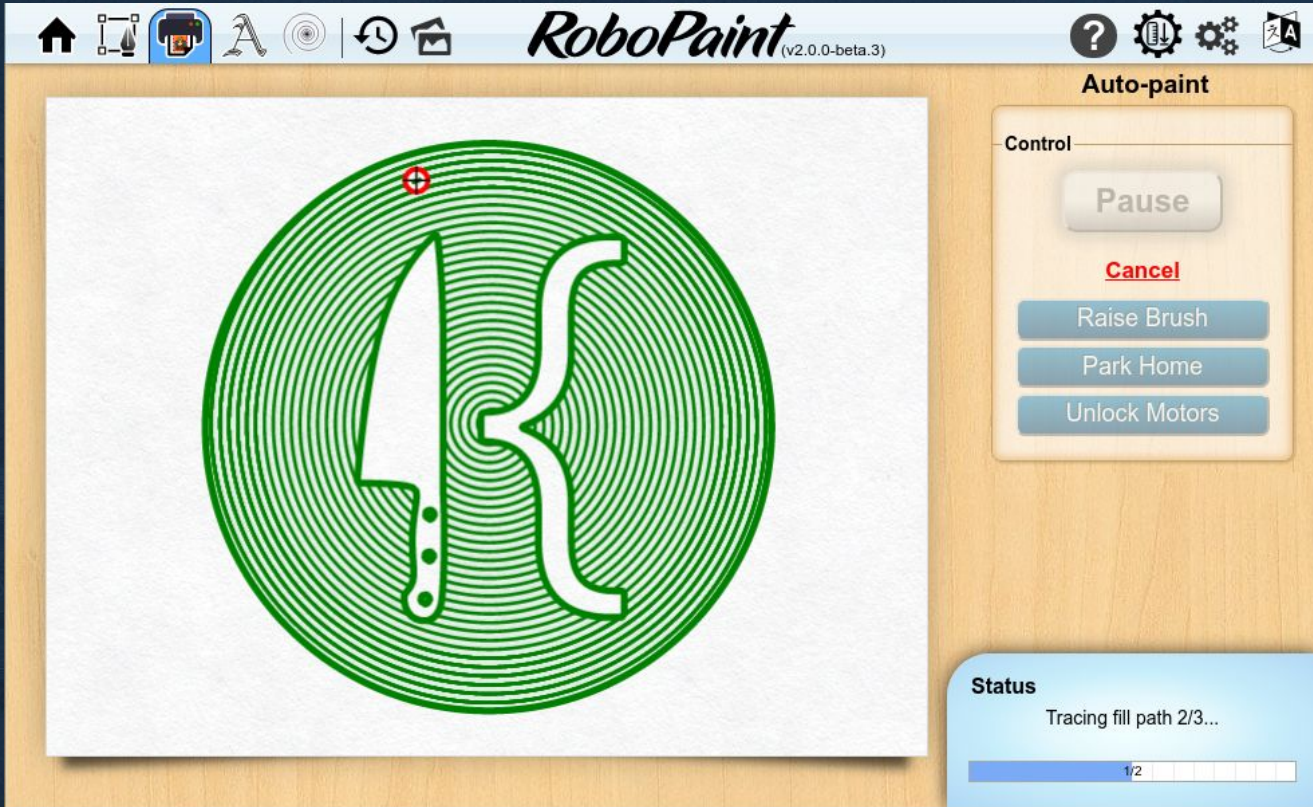
A dark, blue-tinted photograph of a cityscape. In the center, there is a large, multi-story building with a prominent dome and a tall, dark chimney to its right. To the right of the chimney, there is a large, white, tent-like structure with multiple peaks. The foreground shows a body of water reflecting the scene. The text "Just give it a try!" is overlaid in a bold, yellow, serif font, centered horizontally and slightly above the middle vertically.

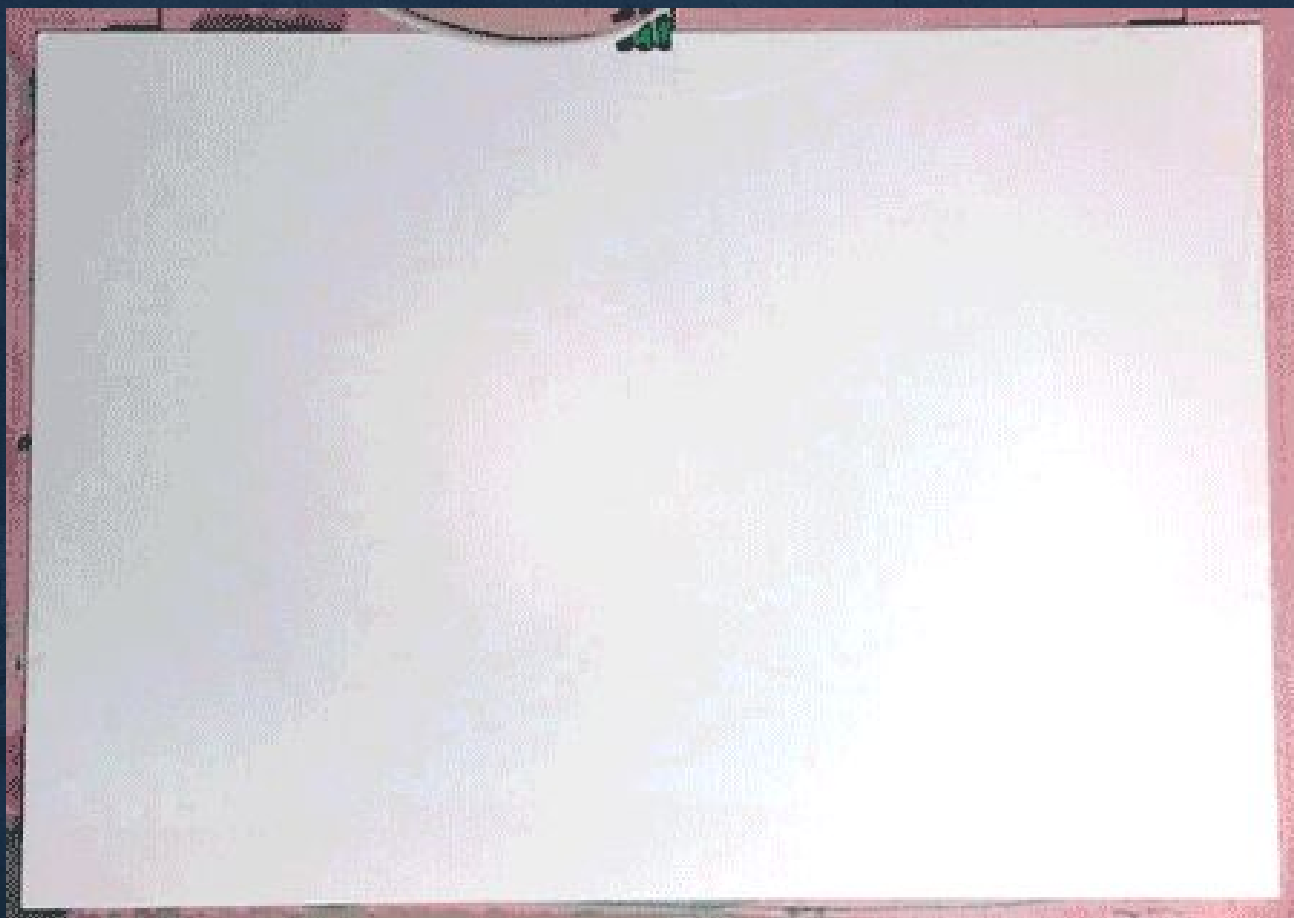
Just give it a try!

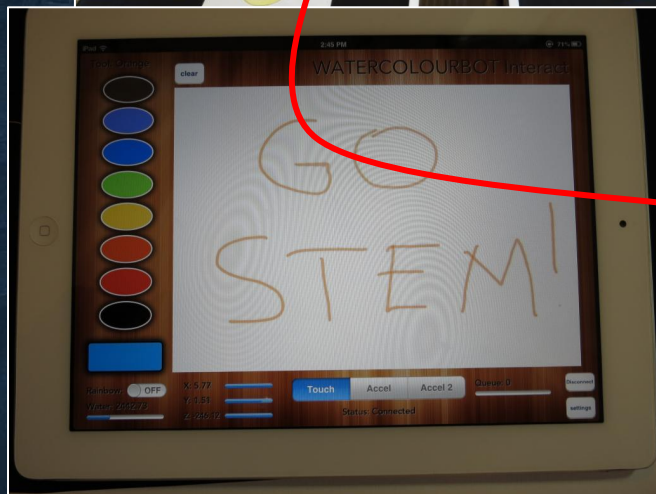
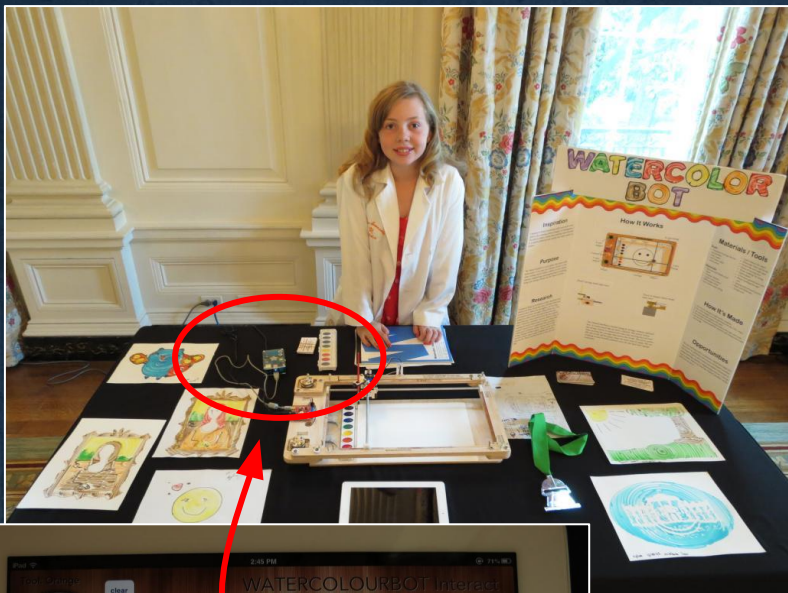
The background is a dark, blue-tinted photograph of a waterfront scene. In the foreground, there's a body of water reflecting the sky. In the middle ground, there's a city skyline with various buildings, including a prominent one with a dome and a tall, dark chimney. To the right, there's a large, white, tent-like structure with multiple peaks. The overall atmosphere is moody and industrial.

“Ship Stuff Every Day”
— Plato

(or was it Snoop Dogg?)

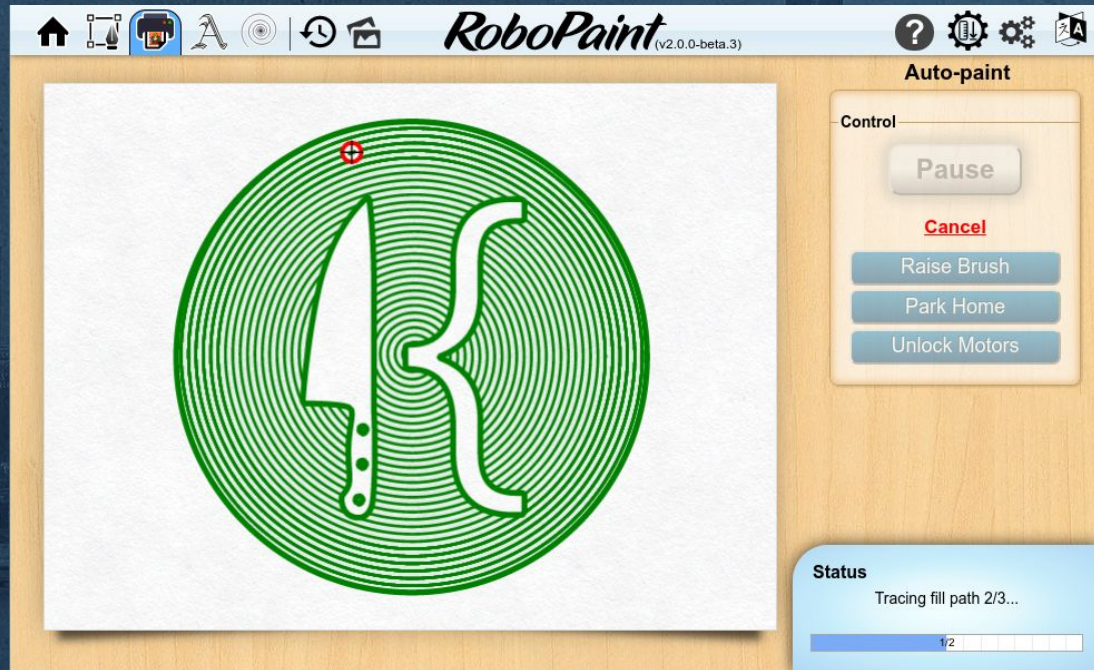






CNCServer running on a
Raspberry PI driving the
WaterColorBot via API from
native iPad app at the
White House Science Fair 2013

<webview>



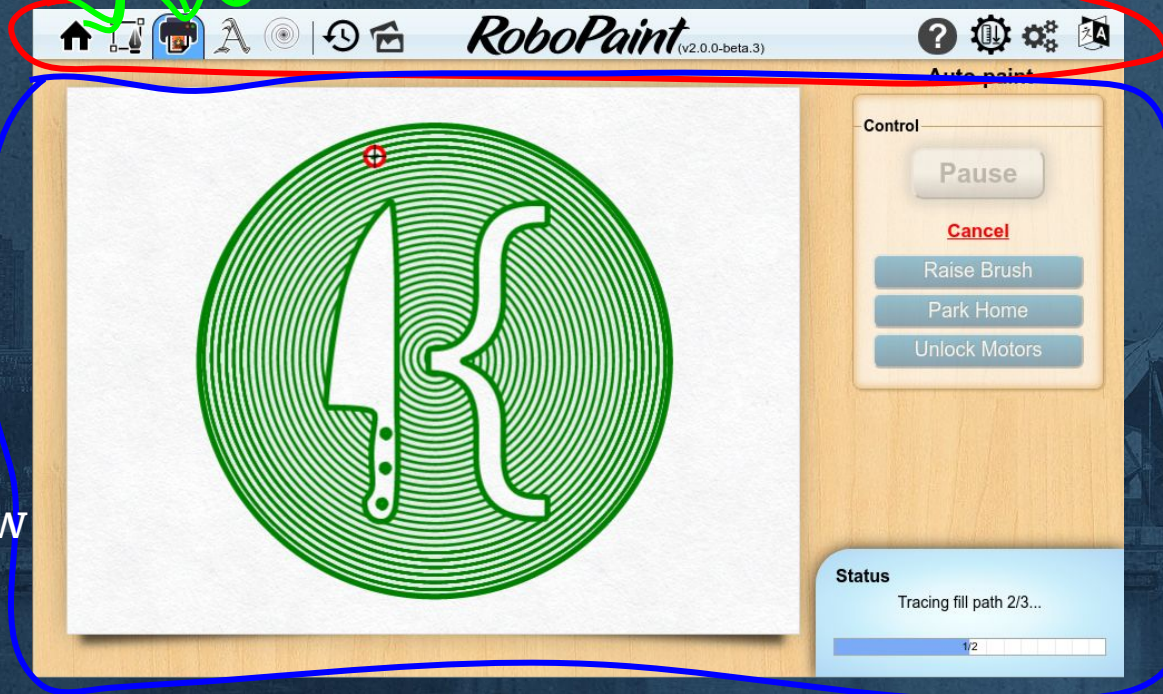
<https://electron.atom.io/docs/api/webview-tag/>

<webview>

Main Window

Mode links

Mode Webview
"iframe"

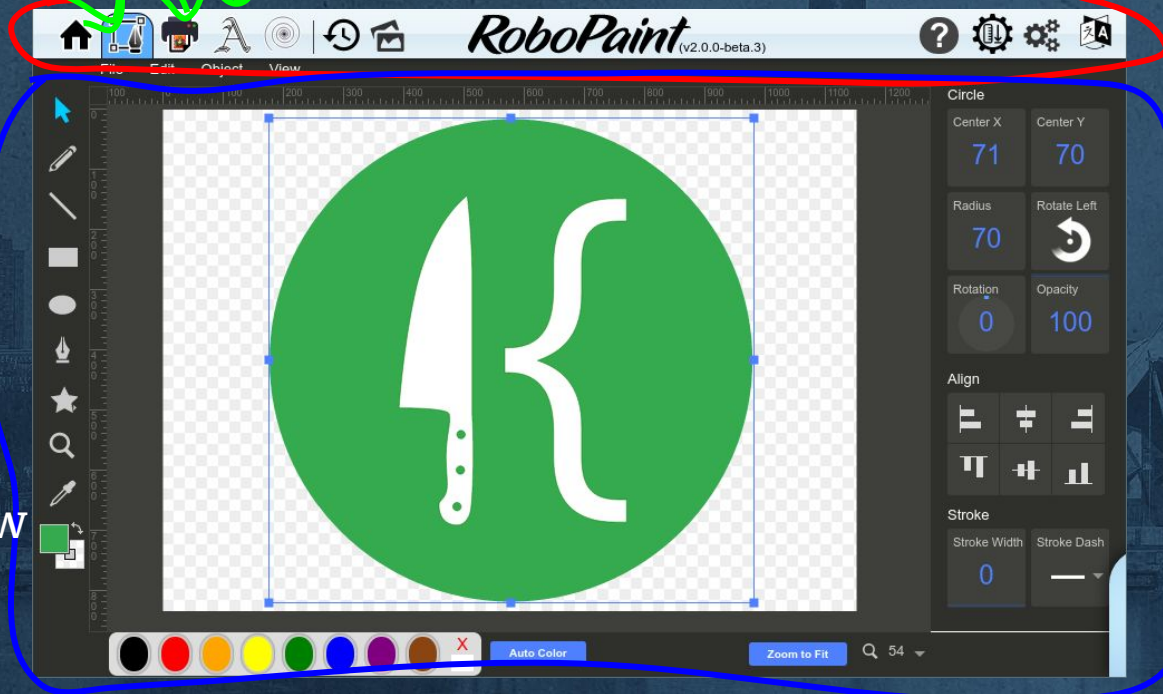


<webview>

Main Window

Mode links

Mode Webview
"iframe"

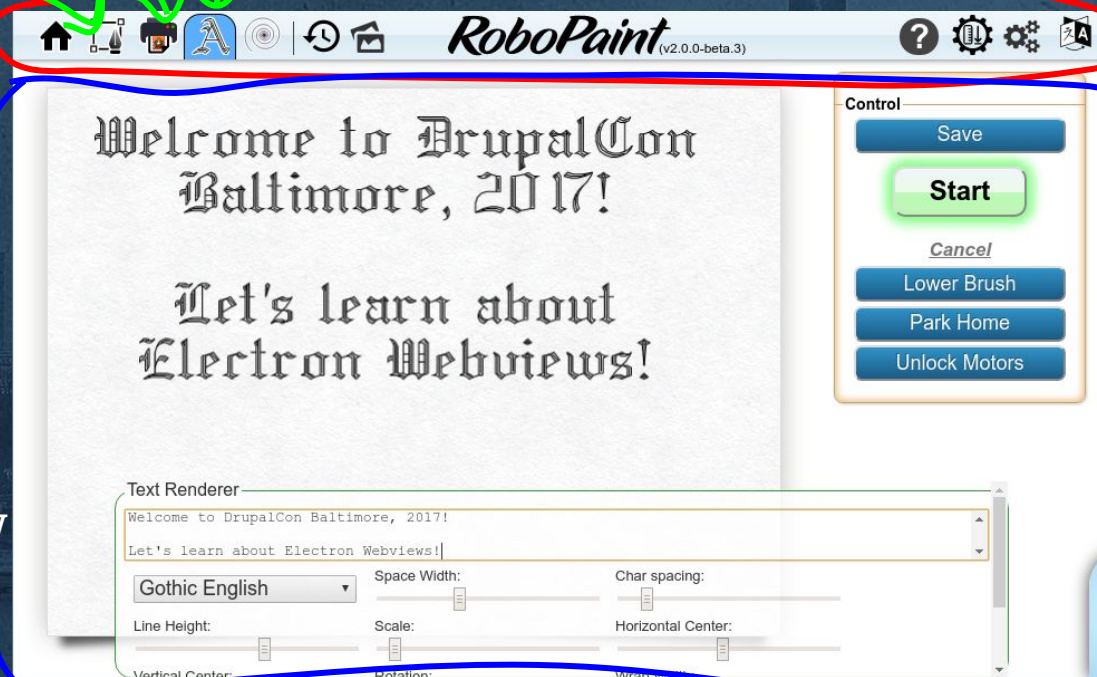


<webview>

Main Window

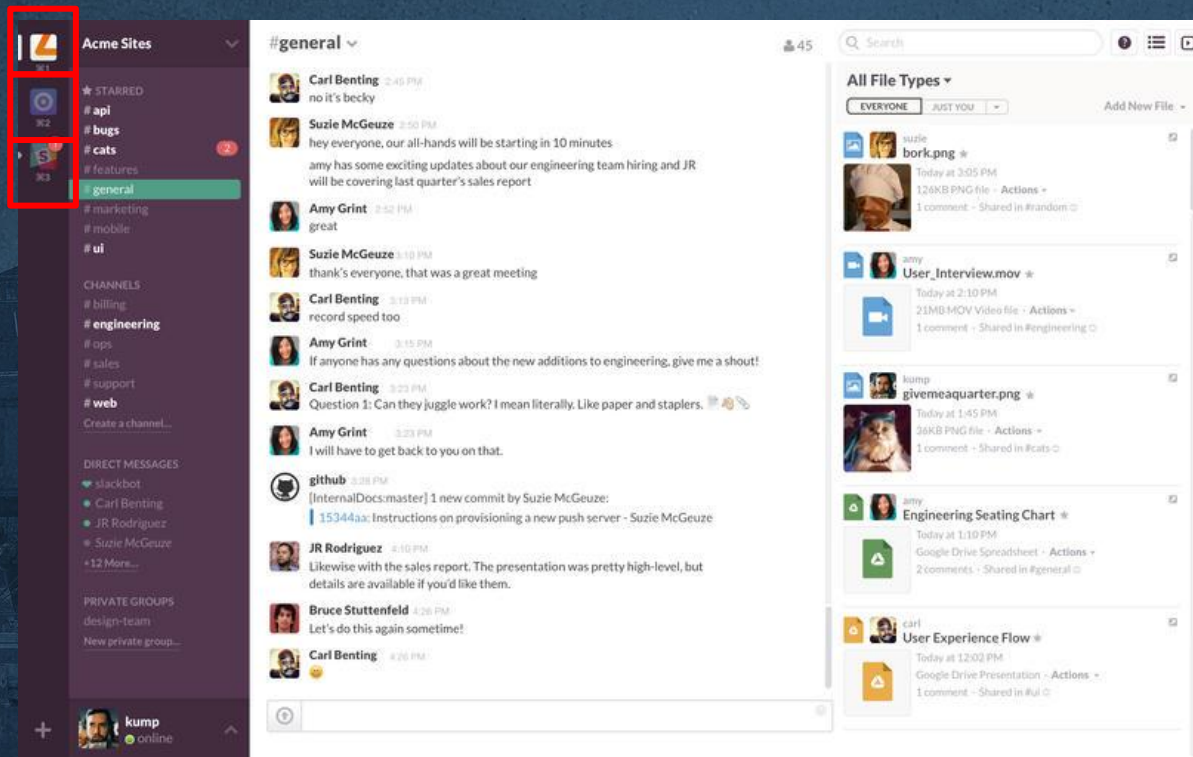
Mode links

Mode Webview
“iframe”



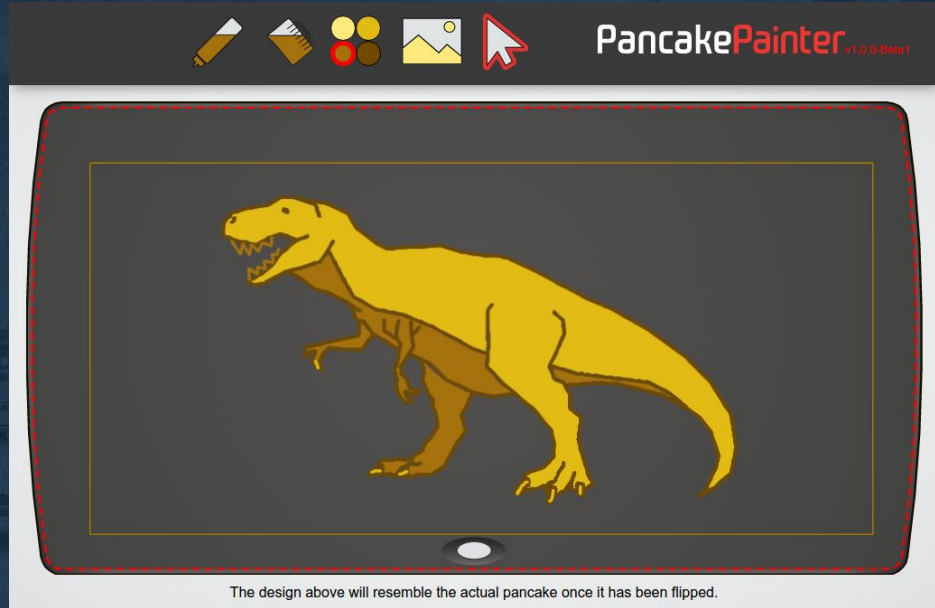
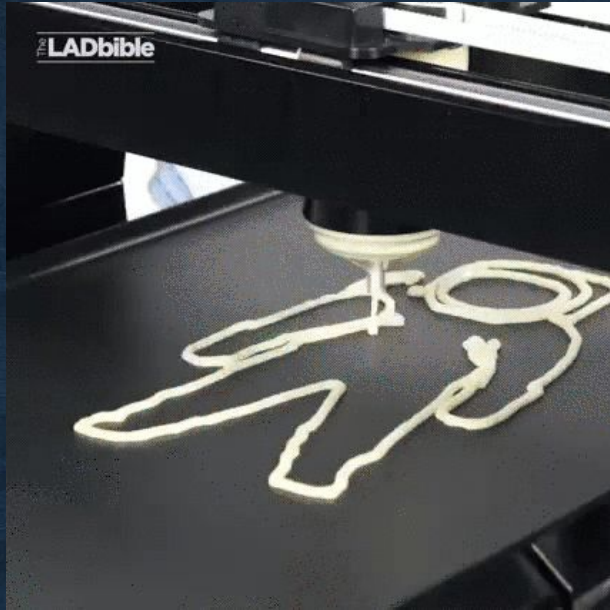
<webview>

Each organization in Slack runs in its own webview process.



<https://slack.engineering/reducing-slacks-memory-footprint-4480fec7e8eb>

PancakeBot™



<https://github.com/PancakeBot/PancakePainter>



Paper.js

<https://github.com/PancakeBot/PancakePainter>



Please check out our [Rules and FAQs](#)



IamA Open Source Developer for the PancakeBot AMA!



TECHNOLOGY submitted 3 months ago * by [techninja42](#)

Yes, it's possibly the most boring AMA in existence, ask a programmer all you ever wanted to know about printing pancakes, or other stuff! :D

It is done! Thank you and happy new year all! I suppose I've answered pretty much every single question over the course of the last 6 hours. I live here so I won't really stop answering questions, but I will go and get some sleep. There's some gems hidden in some of these Q & A's so read up, and thanks for having me. Zoidberg says (/) (°,°,°) (/) *Contribute to Open Source Software!*

My short bio: I'm 33, been programming for 20 something years, son to creator of the almost world famous Underground Comix Company [RipOff Press](#). Got into web development heavily around 2003, fell into programming for robots when my eldest child built a [watercolor painting robot](#) and needed [software for it](#). We then took it around the world, even showed it to [Obama](#). Got noticed by an awesome maker who said he wanted me to make the PancakeBot software, and I said sure! So I made [PancakePainter](#) open source using open web technologies. Fun stuff.

Oh, and I posted that [Adam Savage metaphoto post back in may](#). Good times XD

My Proof: [Twitter Post](#) - [Keybase Proof](#) that I own both twitter and Reddit accounts.

Also check out [ninjanode](#), a fun crappy game I made in a week a few years ago.

1877

267 comments

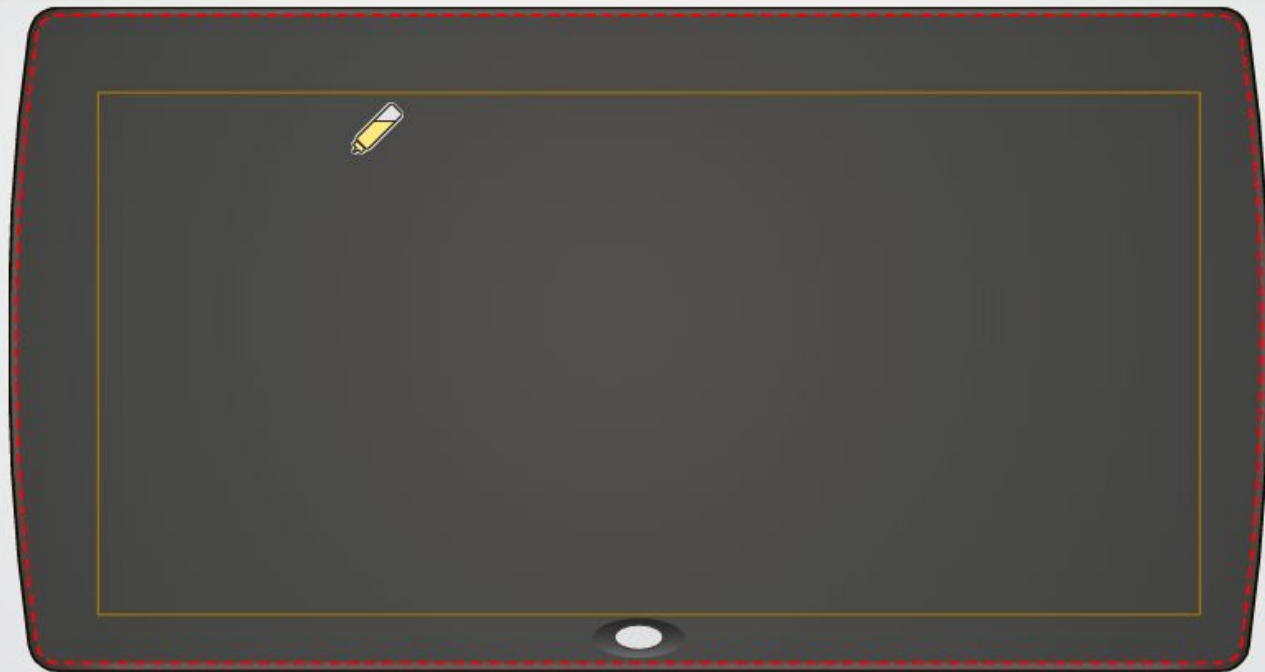
EDIT

share save hide delete nsfw spoiler

<https://redd.it/5ld12n>



PancakePainter v1.3.0-beta

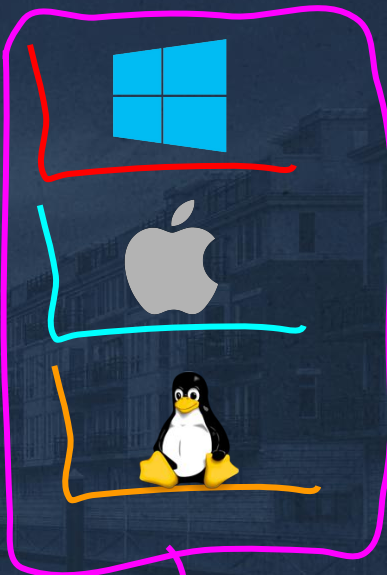


The design above will resemble the actual pancake once it has been flipped.

Raster to autotrace >



OS Specific Binary



< autotrace to SVG



Making Electron Work for you

Let's build a simple Electron App

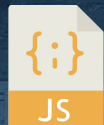
Development Prerequisites:

- Relatively new OS (minimum)
 - Win 7, OS X 10.9, Ubuntu 12
- Terminal/Command line access
- Node.js installed (nodejs.org)
- Working knowledge of how web browsers display HTML

Electron Project Minimums:



package.json



main.js



index.html

package.json

```
{  
  "name": "my-electron-project",  
  "version": "1.0.0",  
  "description": "A minimal Electron application",  
  "main": "main.js",  
  "scripts": {  
    "start": "electron ."  
  },  
  "devDependencies": {  
    "electron": "~1.6.2"  
  }  
}
```



Main.js (Main Process)

```
const electron = require('electron')
const app = electron.app
const BrowserWindow = electron.BrowserWindow

const path = require('path')
const url = require('url')

function createWindow () {
  mainWindow = new BrowserWindow({width: 800, height: 600})

  mainWindow.loadURL(url.format({
    pathname: path.join(__dirname, 'index.html'),
    protocol: 'file:',
    slashes: true
  }))

  app.on('ready', createWindow)
```



Index.html (Renderer Process)

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Hello World!</title>
  </head>
  <body>
    <h1>Hello World!</h1>
    <!-- All of the Node.js APIs are available in this renderer process. -->
    We are using Node.js <script>document.write(process.versions.node)</script>,
    Chromium <script>document.write(process.versions.chrome)</script>,
    and Electron <script>document.write(process.versions.electron)</script>.
  </body>

  <script>
    // You can also require other files to run in this process
    require('./renderer.js')
  </script>
</html>
```



`npm install`, `npm start`

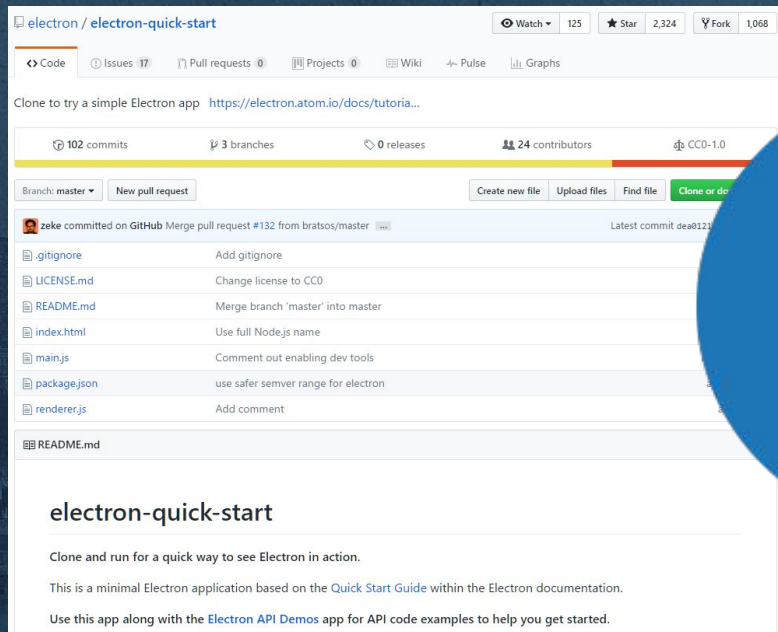
```
techninja@tiger:~/myelectron$
```

It works! (hopefully)

Hello World!

We are using Node.js 7.4.0, Chromium 56.0.2924.87, and Electron 1.6.5.

...or just use electron-quick-start



electron / electron-quick-start

Watch 125 Star 2,324 Fork 1,068

Code Issues 17 Pull requests 0 Projects 0 Wiki Pulse Graphs

Clone to try a simple Electron app <https://electron.atom.io/docs/tutorial...>

102 commits 3 branches 0 releases 24 contributors CC0-1.0

Branch: master New pull request

Create new file Upload files Find file Clone or download

zeke committed on GitHub Merge pull request #132 from bratsos/master Latest commit dea0121

.gitignore	Add gitignore
LICENSE.md	Change license to CC0
README.md	Merge branch 'master' into master
index.html	Use full Nodejs name
main.js	Comment out enabling dev tools
package.json	use safer semver range for electron
renderer.js	Add comment

README.md

electron-quick-start

Clone and run for a quick way to see Electron in action.

This is a minimal Electron application based on the [Quick Start Guide](#) within the Electron documentation.

Use this app along with the [Electron API Demos](#) app for API code examples to help you get started.



```
`git clone https://github.com/electron/electron-quick-start`
```

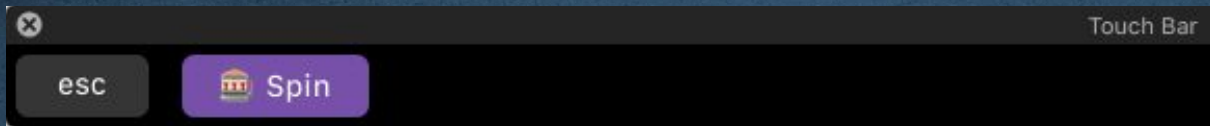
Electron can use...

- Express server
- Native runtimes/compiled code
- Low level hardware connections

Electron Libraries:

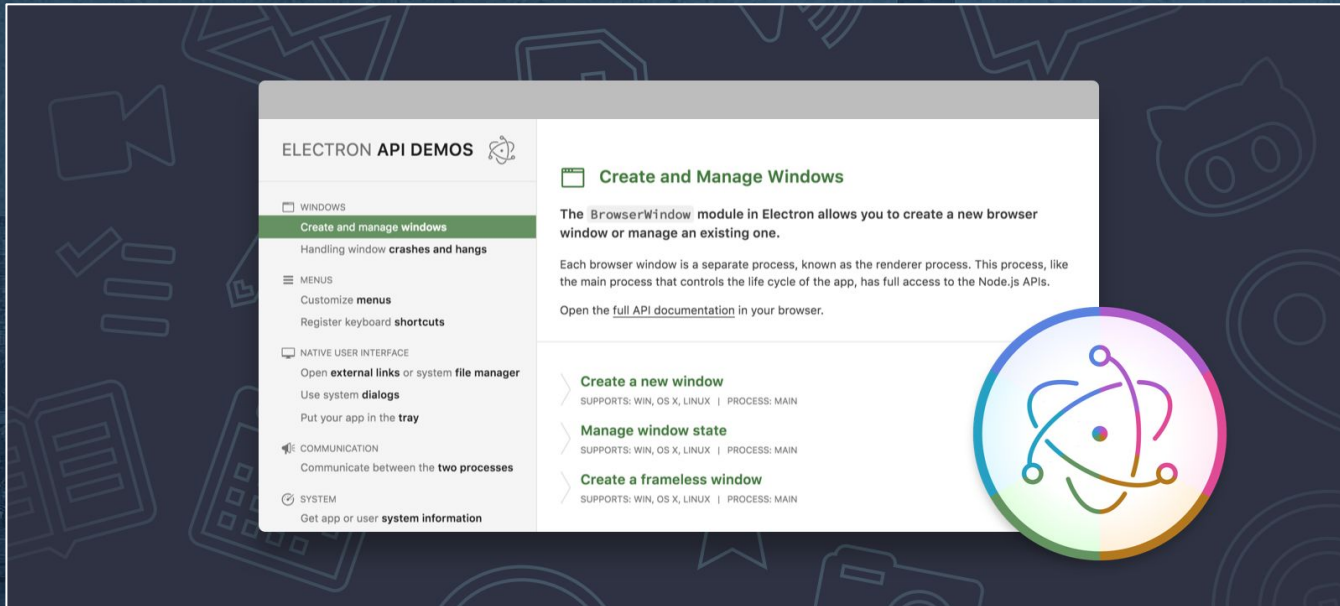
- Automatic babel/sass compilation
- Streamlined user configuration storage
- etc.....

Native APIs



- File Open/Save
- Native customizable dialogs for each OS
- Menus, keyboard shortcuts, clipboard
- Desktop capture
- Environment variables
- IPC communication between processes
- Frameless windows
- Dynamic tray, dock or application icons
- Tons more!

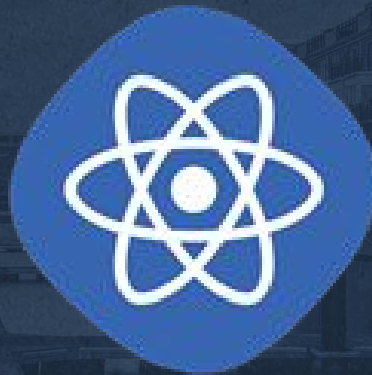
Native API demo app



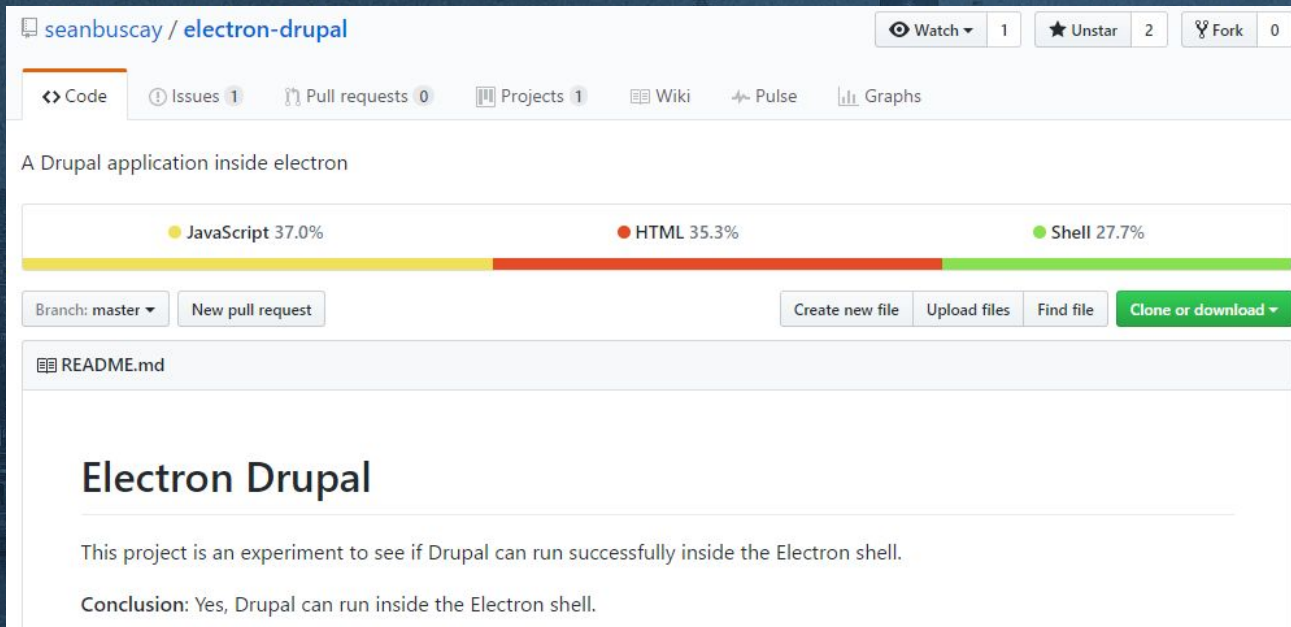
github.com/electron/electron-api-demos

Integration

Drupal 8, Waterwheel, React, and Electron



Drupal 8 *inside* electron



The screenshot shows the GitHub interface for the repository 'seanbuscay / electron-drupal'. At the top, there are buttons for 'Watch' (1), 'Unstar' (2), and 'Fork' (0). Below this is a navigation bar with links for 'Code', 'Issues' (1), 'Pull requests' (0), 'Projects' (1), 'Wiki', 'Pulse', and 'Graphs'. The main content area has the title 'A Drupal application inside electron'. Below the title is a progress bar showing the composition of the application: JavaScript 37.0% (yellow), HTML 35.3% (red), and Shell 27.7% (green). Under the progress bar are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. Below this is a section for the 'README.md' file, which contains the title 'Electron Drupal' and the following text: 'This project is an experiment to see if Drupal can run successfully inside the Electron shell. Conclusion: Yes, Drupal can run inside the Electron shell.'

seanbuscay / electron-drupal

Watch 1 Unstar 2 Fork 0

Code Issues 1 Pull requests 0 Projects 1 Wiki Pulse Graphs

A Drupal application inside electron

JavaScript 37.0% HTML 35.3% Shell 27.7%

Branch: master New pull request Create new file Upload files Find file Clone or download

README.md

Electron Drupal

This project is an experiment to see if Drupal can run successfully inside the Electron shell.

Conclusion: Yes, Drupal can run inside the Electron shell.

README.md

waterwheel-training

4K Four Kitchens

Lessons + Code examples for the 4K Drupal 8 + React + Waterwheel training

Contributors



Prerequisites

- [npm](#) installed
- [yarn](#) for package management
- [Postman](#) for testing out the API
- React Dev Tools ([Chrome](#)/[Firefox](#))
- Redux Dev Tools ([Chrome](#)/[Firefox](#))

Usage

- `git clone git@github.com:fourkitchens/waterwheel-training.git && cd waterwheel-training` - Clone this repo and change directory into it
- `npm use` - Change to supported node version (Node 6) if you have [npm](#) installed

github.com/fourkitchens/waterwheel-training

All the Drupal 8 API/Headless sessions or trainings at DrupalCon Baltimore:

- API First Drupal 8 with React.js and Waterwheel
- Future of the CMS: Decoupled, multichannel, and content-as-a-service
- A look into a possible Future for all of us: React, GraphQL and Drupal
- Ain't No Body: Not Your Mama's Headless Drupal
- Decoupled Drupal and Angular 2
- EmberJS: A Fitting Face for a D8 Backend
- Decoupled from the Inside Out
- Drupal, Alexa, and Big Mouth Billy Bass Walk into a Bar
- API-First Initiative
- Masters of the Universe: Live-coding a React Application using Drupal Services

Main.js (excerpt)

...

```
function createWindow () {  
  mainWindow = new BrowserWindow({width: 740, height: 500})  
  
  mainWindow.loadURL(url.format({  
    pathname: path.join(__dirname, 'build', 'index.html'),  
    protocol: 'file:',  
    slashes: true  
  }))  
  
  app.on('ready', createWindow)  
  
  app.on('window-all-closed', function () {  
    app.quit()  
  })  
}
```

package.json (excerpt)

```
{  
  "name": "todomvc",  
  "version": "0.0.1",  
  "main": "main.js",  
  "scripts": {  
    "start": "electron .",  
    "build": "react-scripts build"  
  },  
  "devDependencies": {  
    "electron": "~1.6.2",  
    "enzyme": "^2.4.1",  
    "react-scripts": "^0.9.0",  
    ...  
  }  
}
```


`npm install`

```
techninja@tiger:~/todo-electron$
```

`npm run build`, `npm start`

```
├── whatwg-fetch@2.0.2
├── redux@3.6.0
├── lodash-es@4.17.4
├── symbol-observable@1.0.4
├── redux-thunk@2.2.0
└── todomvc-app-css@2.1.0

npm WARN optional Skipping failed optional dependency
/react-scripts/fsevents:
npm WARN notsup Not compatible with your operating sy
stem or architecture: fsevents@1.0.17
npm WARN optional Skipping failed optional dependency
/chokidar/fsevents:
npm WARN notsup Not compatible with your operating sy
stem or architecture: fsevents@1.1.1
techninja@tiger:~/todo-electron$
```

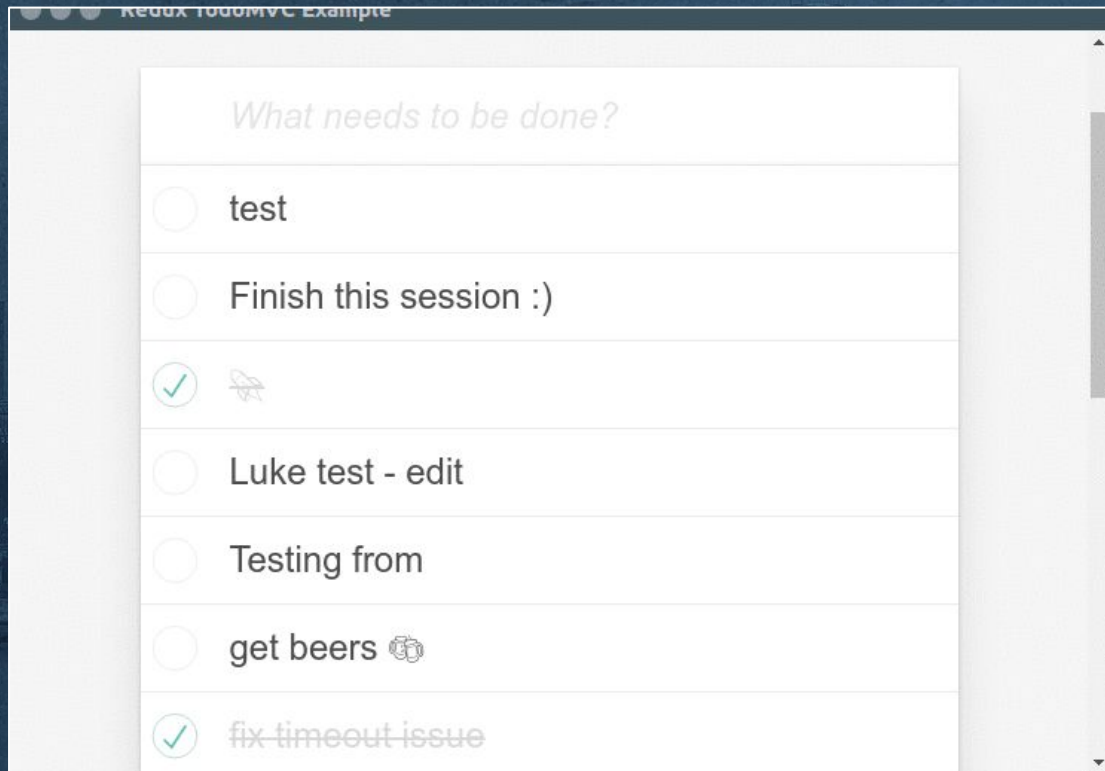
To do React app in Electron!

Redux TodoMVC Example

Please login with your 4k Waterwheel Training credentials.

Log In

API, Devtools for debugging



It's really that easy

The image displays a development environment with three main components:

- Code Editor (main.js):** Shows the main application logic for creating and managing the browser window.
- Terminal:** Shows the command-line execution of the application.
- Browser:** Displays the Redux TodoMVC Example application, which includes a title, a text input, and a list of tasks.

```
let mainWindow

function createWindow () {
  mainWindow = new BrowserWindow({width: 740, height: 500})

  mainWindow.loadURL(url.format({
    pathname: path.join(__dirname, 'build', 'index.html'),
    protocol: 'file:',
    slashes: true
  }))

  mainWindow.on('closed', function () {
    mainWindow = null
  })
}

app.on('ready', createWindow)
```

```
techninja@tiger: ~/todo-electron
> electron .
^C
techninja@tiger:~/todo-electron$ npm start
> todonvc@0.0.1 start /home/techninja/todo-electron
> electron .
^C
techninja@tiger:~/todo-electron$ npm start
> todonvc@0.0.1 start /home/techninja/todo-electron
> electron .
```

The browser window shows the Redux TodoMVC Example application. The title is "todos". The input field contains the text "What needs to be done?". The list of tasks is:

- ☐ get beers 🍺
- ☒ fix timeout issue
- ☒ Like all the Todos →

The browser's developer tools are open, showing the DOM tree and the Styles pane. The DOM tree shows the following structure:

```
<ul class="todo-list">
  <li class="...">
  <li class="completed">
  <li class="completed">
  <li class="...">
  <li class="...">
  <li class="completed">
  <li class="completed">
  <li class="completed">
</ul>
```

The Styles pane shows the following styles for the selected element:

```
position: relative;
font-size: 24px;
border-bottom: 1px solid #ededed;
li {
  display: list-item;
  text-align: -webkit-match-parent;
}
```

Squirrel!

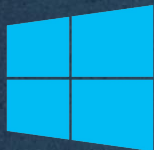


Installation packaging

- Integrating an app should be easy
- Packaging can be automated, and supports delta updates
- Distributing should be straightforward with channels
- Installing is Wizard-Free™ with no UAC dialogs or reboot
- Updating is done in the background



Installation building



electron-packager

electron-installer
windows



appname.exe



electron-packager

appdmg



appname.dmg



electron-packager

electron-installer
[debian/redhat]



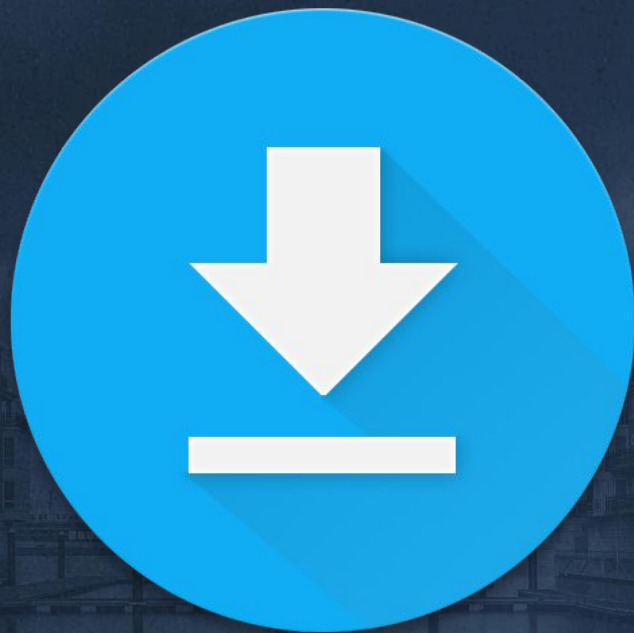
appname.deb/
appname.rpm

**Please wait while
the software for your
PancakeBot is being installed.**

PancakePainter

**Application will launch
when completed.**

ORBO

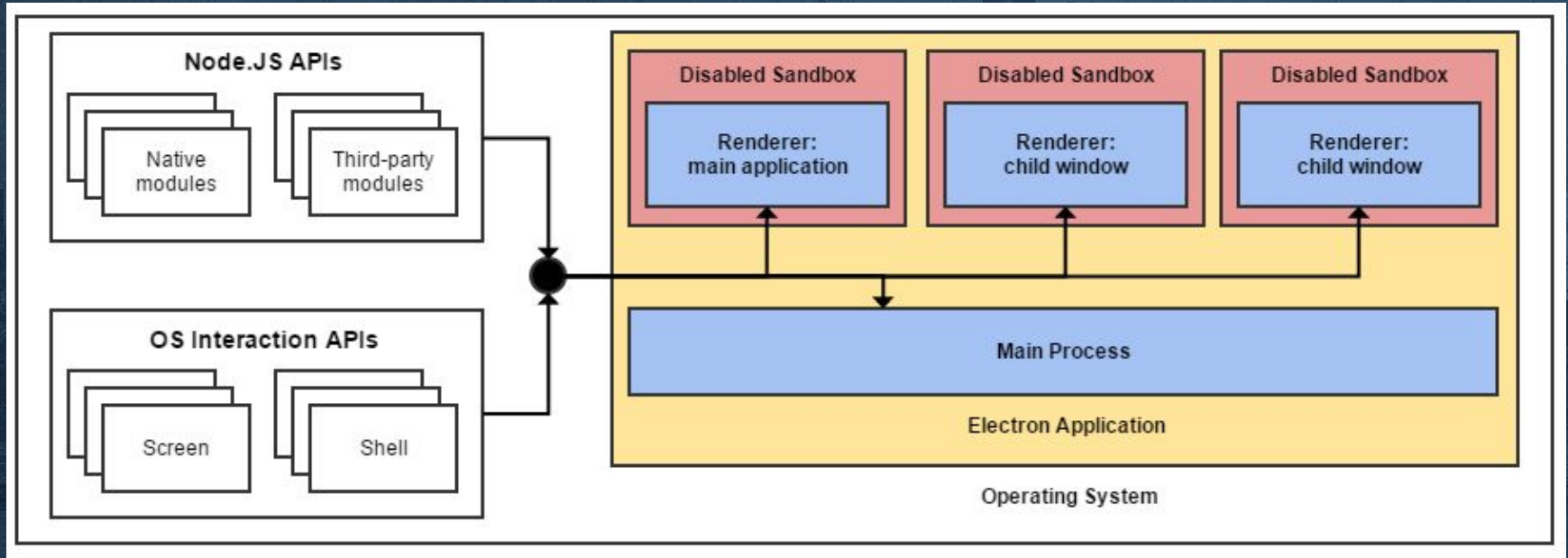


Updates

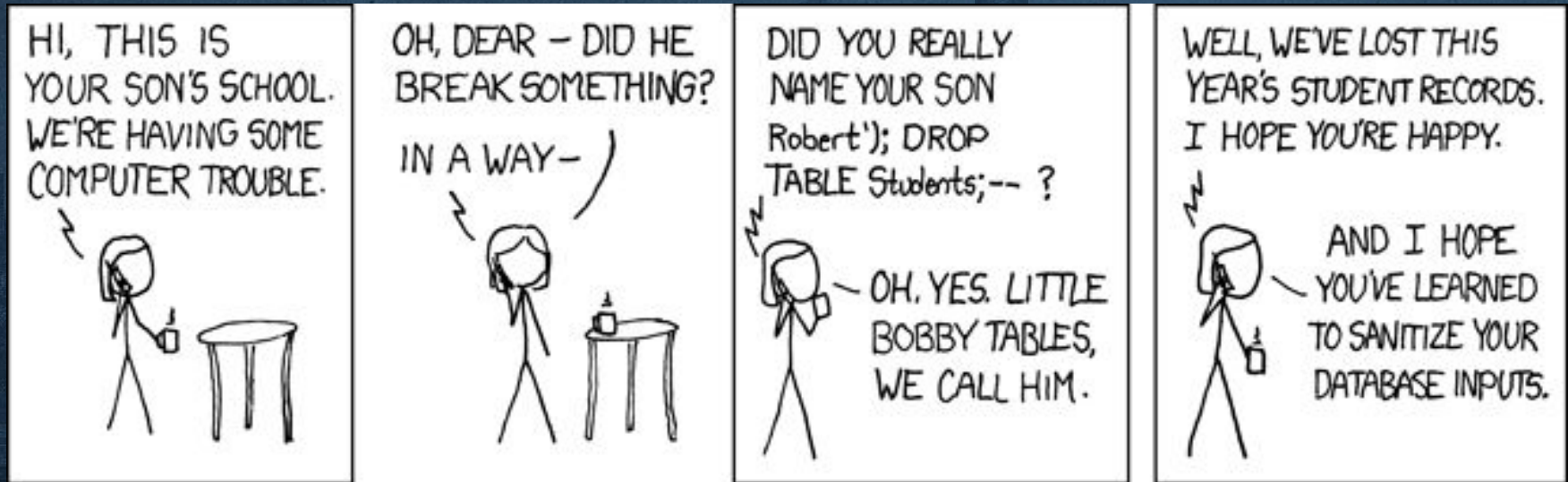
autoUpdater & update servers

- Nuts: <https://github.com/GitbookIO/nuts>
 - Powered by GitHub project releases for files & data
 - Quickly deployable to Heroku/docker/your own server as stateless service with minor configuration via environment variables
 - Supports private repositories for closed source releases
 - Supports Squirrel.Windows & Squirrel.Mac
- “Electron Release Server”:
<https://github.com/ArekSredzki/electron-release-server>
 - Full featured, provides hosted Angular powered UI interface for managing and displaying releases
 - Docker deploy support, or run it on your own server
 - Supports Squirrel.Windows & Squirrel.Mac

Security and sandboxes



Don't let Bobby ruin your week



Electron Security basics

- Always sanitize input and output
- Don't allow giant or executable file uploads
- Only ever load local source files into windows
- Otherwise, disable node integration and use caution

Crash reporting

```
const {crashReporter} = require('electron')

crashReporter.start({
  productName: 'YourName',
  companyName: 'YourCompany',
  submitURL: 'https://your-domain.com/submit',
  uploadToServer: true
})
```



**Mini Breakpad
Server**

github.com/electron/mini-breakpad-server

That's pretty much it!

- Electron is very active and still somewhat new.
- It's not always the right solution
- When in doubt, just give it a try.
- Real companies ship with Electron every day



Electron Question Time!



Thank you!

Join Us for Contribution Sprints

Friday, April 28, 2017

**First-Time Sprinter
Workshop**
9:00am-12:00pm
Room: 307-308

Mentored Core Sprint
9:00am-12:00pm
Room:301-303

General Sprints
9:00am-6:00pm
Room:309-310

#drupalsprints



WHAT DID YOU THINK?

Supercharge Your Next Web App with electron!

<https://events.drupal.org/baltimore2017/sessions/supercharge-your-next-web-app-electron>

Take the survey!

<https://www.surveymonkey.com/r/drupalconbaltimore>

THANK YOU!