Building cross-platform desktop apps using Web Tech

Rahul Rout
Electron
Building cross-platform desktop apps using Web Tech

Rahul Rout
@routbuzz
14\textsuperscript{th} April, 2017
Electron

Framework for building cross-platform desktop applications with web technologies
Mommy, why is he talking about desktop apps??
Here’s Why
GREAT!!!

I STILL DONT KNOW WHY IS HE STILL TALKING ABOUT ELECTRON
Electron Facts
Open Source

When Atom launched, Github open sourced everything including Electron, then called Atom Shell.
Chromium & Node.js

Electron combines Chromium’s rendering library with the full Node.js API into a single runtime with one V8
Chrome

- Use great features like CSS variables
- Use latest chrome releases
Node.js

- The full Node.js API is available in Electron
- File System, Sockets, Child Processes
<script>
    require('./app.js')
</script>
Ship to Three Platforms

- Electron apps work on Mac, Windows and Linux
- Also has an ARM distribution
How Electron Works

- Has two processes which bring an Electron App to life
- Main process
- Renderer Process
Main Process

- The lifecycle of your app, the heavy lifter
  - A JS file which tells Electron what do, which files to open for rendering etc
  - Use it for dialogs etc
Renderer Process

- Simply, the webpage
Electron API Modules

- Browser Window
- Dialog
- Global Shortcut
- Menu
- Power Save Blocker
- Tray
- Desktop Capturer
- ...

@WalmartLabs
API Demos

An Electron app demoing Electron APIs with sample code you can re-use
Create and Manage Windows

The BrowserWindow module in Electron allows you to create a new browser window or manage an existing one.

Each browser window is a separate process, known as the renderer process. This process, like the main process that controls the life cycle of the app, has full access to the Node.js APIs.

Open the [full API documentation](#) in your browser.

- **Create a new window**
  - SUPPORTS: WIN, OS X, LINUX | PROCESS: MAIN

- **Manage window state**
  - SUPPORTS: WIN, OS X, LINUX | PROCESS: MAIN

- **Window events: blur and focus**
  - SUPPORTS: WIN, OS X, LINUX | PROCESS: MAIN

- **Create a frameless window**
  - SUPPORTS: WIN, OS X, LINUX | PROCESS: MAIN
Devtron

An extension to the dev tools which is specific to Electron apps giving information specific to the electron lifecycle
<table>
<thead>
<tr>
<th>Channel</th>
<th>Listeners</th>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>requestShowOpenDialog</td>
<td></td>
<td><code>{{&quot;title&quot;:&quot;Select music files&quot;,&quot;filters&quot;:null}}</code></td>
</tr>
<tr>
<td>finishShowOpenDialog</td>
<td>1</td>
<td><code>[null, null]</code></td>
</tr>
</tbody>
</table>

What does it take to build an app?
package.json
{
  "name": "example-app",
  "version": "1.0.0",
  "description": "",
  "main": "main.js",
  "scripts": {
    "start": "electron ."
  },
  "author": "Rahul Rout",
  "license": "MIT",
  "devDependencies": {
    "electron-prebuilt": "^1.1.0"
  }
}

main.js
const {app, BrowserWindow} = require('electron');
let mainWindow;

app.on('ready', () => {
  mainWindow = new BrowserWindow({
    width: 500,
    height: 400
  });
  mainWindow.loadURL('file:///' + __dirname + '/index.html');
});

index.html
<!DOCTYPE html>
<html>
  <head>
    <title>Example app</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
  </head>
  <body>
    <h1>Hello, world!</h1>
  </body>
</html>
ATOM

A hackable text editor for the 21st Century

For help, please visit:
- The Atom docs for Guides and the API reference.
- The Atom forum at discuss.atom.io.
- The Atom org. This is where all GitHub-created Atom packages can be found.

Note: To help us improve Atom, we anonymously track usage metrics, such as launch time, screen size, and current version. See the atom/metrics package for details and instructions to disable it.

Get to know Atom!

- Open a Project
- Install a Package
- Choose a Theme
- Customize the Styling
- Hack on the Init Script
- Add a Snippet
- Learn some Shortcuts
How do I send it to the world?
Packaging

- Electron-Packager – A node module which helps in bundling your app into a .app (mac) or .exe file
- Electron-Builders
  - Built on top of electron-packager and other modules
  - Creates .dmg and Windows installers
  - Takes care of the code signing for the app
Auto Updates

- Electron has built-in support for automatic updates – autoUpdater API
- Does not support Linux as of now
- Based on Squirrel update framework
  - Implementation differs for Mac and Windows
MOBILE & DISRUPTIVE TECHNOLOGY SUMMIT

October 5-6, 2017
Indian Institute of Science, Bangalore
www.modsummit.com

Register early and get the best discounts

GREAT INDIAN DEVELOPER SUMMIT

April 23-28, 2018
Indian Institute of Science, Bangalore
www.developersummit.com