



building an electron-based application.

One of the things I've always wondered is how to package web content to make it really independent of the browser we're rendering it in.

The first time I saw this was in [Adobe Brackets](#), at the time my favorite text editor. It used a highly customized version of the [Chrome Embedded Framework](#) that appeared to offer a lot of flexibility in building your own applications on top of web technologies. However Adobe made that harder:

Note: The brackets-shell is only maintained for use by the Brackets project. Although some people have definitely had success using it as an app shell for other projects, we don't provide any official support for that and we haven't done a ton of work to make the app shell easily reusable. Many people will likely find it easier to use a project like NW.js or electron, which is more generic by design.

<https://github.com/adobe/brackets-shell>

So I did what they suggested and started looking at [nw.js](#) and [electron](#). They both leverage Chromium, the open source codebase Chrome is based on, but there are some differences that make me lean towards electron. It's what the [Atom](#) editor is built on and it is maintained by Github and it allows creation of Windows, Linux and Macintosh native applications that are ready for signing and, hopefull, release to their respective stores.

The idea is to build a native application that will act as book reader. As a MVP we want the application to work with bundled content; in later iterations we can load additional bundles, each corresponding to a new book or a new issue of a magazine.

Constraints

Currently, electron is bundled with Chromium 61. This version may be too old for some of the layout techniques I want to use. It may become an issue with upgrading the bundled Chromium, if possible, or changing expectations.

Getting Started

Because we're creating a Node-based application we'll go with the standard node set up and configuration. In this stage we'll perform the following actions:

1. Create package.json
2. Install and Initialize eslint with Google's preset
3. Add .editorconfig
4. Install Gulp globally and locally
5. Install the latest, non beta, version of electron

{{{ Put a screenflow short for the steps in the process }}}}

This will get us ready to plan and build the application.

Planning the app.

The first thing we need to do when designing an application for Electron application is to plan what it will look like and whether it makes sense to package it in electron.

Links and Resources

- [Chromium Embedded Framework](#)
- [Brackets Shell](#)
- [nw.js](#)
- [nw.js Github repo](#)
- [electron](#)
- [Electron Simple Samples](#)
- [Electron Sample Apps](#)
- [Awesome Electron](#)
- [electron-builder](#)