**Electron Builder**

Link : https://www.electron.build/

Installation

Yarn is strongly recommended instead of npm.

yarn add electron-builder --dev

Platform specific 7zip-bin-\* packages are optionalDependencies, which may require manual install if you have npm configured to not install optional deps by default.

Boilerplates

• electron-webpack-quick-start — A bare minimum project structure to get started developing with electron-webpack. Recommended.

• electron-react-boilerplate A boilerplate for scalable cross-platform desktop apps.

• electron-react-redux-boilerplate A minimal boilerplate to get started with Electron, React and Redux.

• electron-boilerplate A minimalistic yet comprehensive boilerplate application.

Quick Setup Guide

electron-webpack-quick-start is a recommended way to create a new Electron application.

• Specify the standard fields in the application package.json — name, description, version and author.

• Specify the build configuration in the package.json as follows:

"build": {  
 "appId": "your.id",  
 "mac": {  
 "category": "your.app.category.type"  
 }  
 }

• See all options.

• Add icons.

• Add the scripts key to the development package.json

• "scripts": {  
 "pack": "electron-builder --dir",  
 "dist": "electron-builder"  
 }

• Then you can run yarn dist (to package in a distributable format (e.g. dmg, windows installer, deb package)) or yarn pack (only generates the package directory without really packaging it. This is useful for testing purposes).

• To ensure your native dependencies are always matched electron version, simply add script "postinstall": "electron-builder install-app-deps" to your package.json.

• If you have native addons of your own that are part of the application (not as a dependency), set nodeGypRebuild to true.

• Install the required system packages if you are not on macOS 10.12+.

Please note that everything is packaged into an asar archive by default.