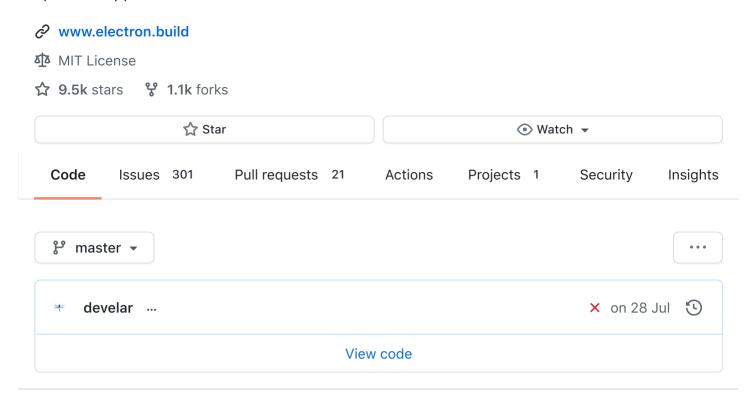
☐ electron-userland / electron-builder

A complete solution to package and build a ready for distribution Electron app with "auto update" support out of the box



README.md



A complete solution to package and build a ready for distribution Electron, Proton Native app for macOS, Windows and Linux with "auto update" support out of the box.

See documentation on electron.build.

- NPM packages management:
 - Native application dependencies compilation (including Yarn support).
 - Development dependencies are never included. You don't need to ignore them explicitly.
 - Two package.json structure is supported, but you are not forced to use it even if you have native production dependencies.
- Code Signing on a CI server or development machine.
- Auto Update ready application packaging.
- Numerous target formats:

- All platforms: 7z, zip, tar.xz, tar.7z, tar.lz, tar.gz, tar.bz2, dir (unpacked directory).
- o macOS: dmg , pkg , mas .
- Linux: Applmage, snap, debian package (deb), rpm, freebsd, pacman, p5p,
 apk.
- Windows: nsis (Installer), nsis-web (Web installer), portable (portable app without installation), AppX (Windows Store), MSI, Squirrel.Windows.
- Publishing artifacts to GitHub Releases, Amazon S3, DigitalOcean Spaces and Bintray.
- Advanced building:
 - Pack in a distributable format already packaged app.
 - Separate build steps.
 - Build and publish in parallel, using hard links on CI server to reduce IO and disk space usage.
 - electron-compile support (compile for release-time on the fly on build).
- Docker images to build Electron app for Linux or Windows on any platform.
- Proton Native support.
- Downloads all required tools files on demand automatically (e.g. to code sign windows application, to make AppX), no need to setup.

Question	Answer
"I want to configure electron-builder"	See options
"I have a question"	Open an issue or join the chat
"I found a bug"	Open an issue
"I want to support development"	Donate

Installation

Yarn is strongly recommended instead of npm.

yarn add electron-builder --dev

Quick Setup Guide

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

 Specify the standard fields in the application package.json — name, description, version and author. 2. Specify the build configuration in the package. ison as follows:

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

See all options. Option files to indicate which files should be packed in the final application, including the entry file, maybe required.

- 3. Add icons.
- 4. 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.

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

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

For an app that will be shipped to production, you should sign your application. See Where to buy code signing certificates.

Donate

We do this open source work in our free time. If you'd like us to invest more time on it, please donate. Donation can be used to increase some issue priority.

Sponsors



Releases 705



+ 704 releases

Sponsor this project



Learn more about GitHub Sponsors

Packages

No packages published

Used by 5,000+



Contributors 299



+ 288 contributors

Languages

- TypeScript 85.7% NSIS 6.8% JavaScript 6.6% Shell 0.4% Dockerfile 0.3%
- HTML 0.2%