



circleci passing build passing chat 1459 online

:memo: Available Translations: CN BR ES JP RU FR US DE. View these docs in other languages at [electron/i18n](https://electronjs.org/i18n).

The Electron framework lets you write cross-platform desktop applications using JavaScript, HTML and CSS. It is based on [Node.js](https://nodejs.org/) and [Chromium](https://www.chromium.org/) and is used by the [Atom editor](https://atom.io/) and many other [apps](#).

Follow [@ElectronJS](https://twitter.com/ElectronJS) on Twitter for important announcements.

This project adheres to the Contributor Covenant [code of conduct](https://github.com/electron/electron/blob/master/CONTRIBUTING.md#code-of-conduct). By participating, you are expected to uphold this code. Please report unacceptable behavior to [coc@electronjs.org](mailto:coc@electronjs.org).

## Installation

To install prebuilt Electron binaries, use [npm](#). The preferred method is to install Electron as a development dependency in your app:

```
npm install electron --save-dev
```

For more installation options and troubleshooting tips, see [installation](#). For info on how to manage Electron versions in your apps, see [Electron versioning](#).

## Platform support

Each Electron release provides binaries for macOS, Windows, and Linux.

- macOS (El Capitan and up): Electron provides 64-bit Intel and ARM binaries for macOS. Apple Silicon support was added in Electron 11.
- Windows (Windows 7 and up): Electron provides `ia32` ( `x86` ), `x64` ( `amd64` ), and `arm64` binaries for Windows. Windows on ARM support was added in Electron 5.0.8.
- Linux: The prebuilt binaries of Electron are built on Ubuntu 20.04. They have also been verified to work on:
  - Ubuntu 14.04 and newer
  - Fedora 24 and newer
  - Debian 8 and newer

## Quick start & Electron Fiddle

Use [Electron Fiddle](#) to build, run, and package small Electron experiments, to see code examples for all of Electron's APIs, and to try out different versions of Electron. It's designed to make the start of your journey with Electron easier.

Alternatively, clone and run the [electron/electron-quick-start](https://github.com/electron/electron-quick-start) repository to see a minimal Electron app in action:

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

## Resources for learning Electron

- [electronjs.org/docs](https://electronjs.org/docs) - All of Electron's documentation
- [electron/fiddle](https://electron/fiddle) - A tool to build, run, and package small Electron experiments
- [electron/electron-quick-start](https://electron/electron-quick-start) - A very basic starter Electron app
- [electronjs.org/community#boilerplates](https://electronjs.org/community#boilerplates) - Sample starter apps created by the community

## Programmatic usage

Most people use Electron from the command line, but if you require `electron` inside your **Node app** (not your Electron app) it will return the file path to the binary. Use this to spawn Electron from Node scripts:

```
const electron = require('electron')
const proc = require('child_process')

// will print something similar to /Users/maf/.../Electron
console.log(electron)

// spawn Electron
const child = proc.spawn(electron)
```

## Mirrors

- [China](#)

See the [Advanced Installation Instructions](#) to learn how to use a custom mirror.

## Documentation translations

We crowdsource translations for our documentation via [Crowdin](#). We currently accept translations for Chinese (Simplified), French, German, Japanese, Portuguese, Russian, and Spanish.

## Contributing

If you are interested in reporting/fixing issues and contributing directly to the code base, please see [CONTRIBUTING.md](#) for more information on what we're looking for and how to get started.

## Community

Info on reporting bugs, getting help, finding third-party tools and sample apps, and more can be found on the [Community page](#).

## License

[MIT](#)

When using Electron logos, make sure to follow [OpenJS Foundation Trademark Policy](#).