

Source Code Directory Structure

The source code of Electron is separated into a few parts, mostly following Chromium on the separation conventions.

You may need to become familiar with [Chromium's multi-process architecture](#) to understand the source code better.

Structure of Source Code

```
Electron
├─ build/ - Build configuration files needed to build with GN.
├─ buildflags/ - Determines the set of features that can be conditionally built.
├─ chromium_src/ - Source code copied from Chromium that isn't part of the content
layer.
├─ default_app/ - A default app run when Electron is started without
|                 providing a consumer app.
├─ docs/ - Electron's documentation.
|   ├─ api/ - Documentation for Electron's externally-facing modules and APIs.
|   ├─ development/ - Documentation to aid in developing for and with Electron.
|   ├─ fiddles/ - A set of code snippets one can run in Electron Fiddle.
|   ├─ images/ - Images used in documentation.
|   └─ tutorial/ - Tutorial documents for various aspects of Electron.
├─ lib/ - JavaScript/TypeScript source code.
|   ├─ browser/ - Main process initialization code.
|   |   ├─ api/ - API implementation for main process modules.
|   |   └─ remote/ - Code related to the remote module as it is
|   |               used in the main process.
|   ├─ common/ - Relating to logic needed by both main and renderer processes.
|   |   └─ api/ - API implementation for modules that can be used in
|   |           both the main and renderer processes
|   ├─ isolated_renderer/ - Handles creation of isolated renderer processes when
|   |                       contextIsolation is enabled.
|   ├─ renderer/ - Renderer process initialization code.
|   |   ├─ api/ - API implementation for renderer process modules.
|   |   ├─ extension/ - Code related to use of Chrome Extensions
|   |   |               in Electron's renderer process.
|   |   ├─ remote/ - Logic that handles use of the remote module in
|   |   |           the main process.
|   |   └─ web-view/ - Logic that handles the use of webviews in the
|   |                 renderer process.
|   ├─ sandboxed_renderer/ - Logic that handles creation of sandboxed renderer
|   |                       processes.
|   |   └─ api/ - API implementation for sandboxed renderer processes.
|   └─ worker/ - Logic that handles proper functionality of Node.js
|               environments in Web Workers.
├─ patches/ - Patches applied on top of Electron's core dependencies
|   |         in order to handle differences between our use cases and
|   |         default functionality.
|   ├─ boringssl/ - Patches applied to Google's fork of OpenSSL, BoringSSL.
|   ├─ chromium/ - Patches applied to Chromium.
|   └─ node/ - Patches applied on top of Node.js.
```

```

|   └─ v8/ - Patches applied on top of Google's V8 engine.
└─ shell/ - C++ source code.
|   └─ app/ - System entry code.
|   └─ browser/ - The frontend including the main window, UI, and all of the
|       |           main process things. This talks to the renderer to manage web
|       |           pages.
|       └─ ui/ - Implementation of UI stuff for different platforms.
|           └─ cocoa/ - Cocoa specific source code.
|           └─ win/ - Windows GUI specific source code.
|           └─ x/ - X11 specific source code.
|       └─ api/ - The implementation of the main process APIs.
|       └─ net/ - Network related code.
|       └─ mac/ - Mac specific Objective-C source code.
|       └─ resources/ - Icons, platform-dependent files, etc.
└─ renderer/ - Code that runs in renderer process.
    └─ api/ - The implementation of renderer process APIs.
└─ common/ - Code that used by both the main and renderer processes,
    |           including some utility functions and code to integrate node's
    |           message loop into Chromium's message loop.
    └─ api/ - The implementation of common APIs, and foundations of
        |           Electron's built-in modules.
└─ spec/ - Components of Electron's test suite run in the renderer process.
└─ spec-main/ - Components of Electron's test suite run in the main process.
└─ BUILD.gn - Building rules of Electron.

```

Structure of Other Directories

- **.circleci** - Config file for CI with CircleCI.
- **.github** - GitHub-specific config files including issues templates and CODEOWNERS.
- **dist** - Temporary directory created by `script/create-dist.py` script when creating a distribution.
- **node_modules** - Third party node modules used for building.
- **npm** - Logic for installation of Electron via npm.
- **out** - Temporary output directory of `ninja`.
- **script** - Scripts used for development purpose like building, packaging, testing, etc.

```

script/ - The set of all scripts Electron runs for a variety of purposes.
└─ codesign/ - Fakes codesigning for Electron apps; used for testing.
└─ lib/ - Miscellaneous python utility scripts.
└─ release/ - Scripts run during Electron's release process.
    └─ notes/ - Generates release notes for new Electron versions.
    └─ uploaders/ - Uploads various release-related files during release.

```

- **typings** - TypeScript typings for Electron's internal code.