'webpack' 不是内部或外部命令，也不是可运行的程序

或批处理文件。

导航

# **Installation**

This guide goes through the various methods used to install webpack.

## **Prerequisites**

Before we begin, make sure you have a **fresh** version of [Node.js](https://nodejs.org/en/) installed. The current **Long Term Support** (LTS) release is an ideal starting point. You may run into a variety of issues with the older versions as they may be missing functionality webpack and/or its related packages require.

## **Local Installation**

The latest webpack release is:

[IMG_256](https://github.com/webpack/webpack/releases)

To install the latest release or a specific version, run one of the following commands:

npm install --save-dev webpack

# or specific version

npm install --save-dev webpack@<version>

**Tip**

Whether to use --save-dev or not depends on your use cases. Say you're using webpack only for bundling, then it's suggested that you install it with --save-dev option since you're not going to include webpack in your production build. Otherwise you can ignore --save-dev.

If you're using webpack v4 or later and want to call webpack from the command line, you'll also need to install the [CLI](https://webpack.js.org/api/cli/).

npm install --save-dev webpack-cli

Installing locally is what we recommend for most projects. This makes it easier to upgrade projects individually when breaking changes are introduced. Typically webpack is run via one or more [npm scripts](https://docs.npmjs.com/misc/scripts) which will look for a webpack installation in your local node\_modules directory:

"scripts": {

"build": "webpack --config webpack.config.js"}

**Tip**

To run the local installation of webpack you can access its binary version as node\_modules/.bin/webpack. Alternatively, if you are using npm v5.2.0 or greater, you can run npx webpack to do it.

## **Global Installation**

The following NPM installation will make webpack available globally:

npm install --global webpack

**Warning**

Note that this is ****not a recommended practice****. Installing globally locks you down to a specific version of webpack and could fail in projects that use a different version.

## **Bleeding Edge**

If you are enthusiastic about using the latest that webpack has to offer, you can install beta versions or even directly from the webpack repository using the following commands:

npm install --save-dev webpack@next# or a specific tag/branchnpm install --save-dev webpack/webpack#<tagname/branchname>

**Warning**

Take caution when installing these bleeding edge releases! They may still contain bugs and therefore should not be used in production.

[Edit this page](https://github.com/webpack/webpack.js.org/edit/master/src/content/guides/installation.mdx)**·**Print this page

# **Introduction**

This boilerplate is targeted towards large, serious projects and assumes you are somewhat familiar with Webpack and vue-loader. Make sure to also read [vue-loader's documentation](https://vue-loader.vuejs.org/" \t "http://vuejs-templates.github.io/webpack/_blank) for common workflow recipes.

If you just want to try out vue-loader or whip out a quick prototype, use the [webpack-simple](https://github.com/vuejs-templates/webpack-simple" \t "http://vuejs-templates.github.io/webpack/_blank) template instead.

## **Quickstart**

To use this template, scaffold a project with [vue-cli](https://github.com/vuejs/vue-cli" \t "http://vuejs-templates.github.io/webpack/_blank). ****It is recommended to use npm 3+ for a more efficient dependency tree.****

$ npm install -g vue-cli

$ vue init webpack my-project

$ cd my-project

$ npm install

$ npm run dev

# **Project Structure**

.

├── build/ # webpack config files

│ └── ...

├── config/

│ ├── index.js # main project config

│ └── ...

├── src/

│ ├── main.js # app entry file

│ ├── App.vue # main app component

│ ├── components/ # ui components

│ │ └── ...

│ └── assets/ # module assets (processed by webpack)

│ └── ...

├── static/ # pure static assets (directly copied)

├── test/

│ └── unit/ # unit tests

│ │ ├── specs/ # test spec files

│ │ ├── eslintrc # config file for eslint with extra settings only for unit tests

│ │ ├── index.js # test build entry file

│ │ ├── jest.conf.js # Config file when using Jest for unit tests

│ │ └── karma.conf.js # test runner config file when using Karma for unit tests

│ │ ├── setup.js # file that runs before Jest runs your unit tests

│ └── e2e/ # e2e tests

│ │ ├── specs/ # test spec files

│ │ ├── custom-assertions/ # custom assertions for e2e tests

│ │ ├── runner.js # test runner script

│ │ └── nightwatch.conf.js # test runner config file

├── .babelrc # babel config

├── .editorconfig # indentation, spaces/tabs and similar settings for your editor

├── .eslintrc.js # eslint config

├── .eslintignore # eslint ignore rules

├── .gitignore # sensible defaults for gitignore

├── .postcssrc.js # postcss config

├── index.html # index.html template

├── package.json # build scripts and dependencies

└── README.md # Default README file

### **build/**

This directory holds the actual configurations for both the development server and the production webpack build. Normally you don't need to touch these files unless you want to customize Webpack loaders, in which case you should probably look at build/webpack.base.conf.js.

### **config/index.js**

This is the main configuration file that exposes some of the most common configuration options for the build setup. See [API Proxying During Development](http://vuejs-templates.github.io/webpack/proxy.html) and [Integrating with Backend Framework](http://vuejs-templates.github.io/webpack/backend.html) for more details.

### **src/**

This is where most of your application code will live in. How to structure everything inside this directory is largely up to you; if you are using Vuex, you can consult the [recommendations for Vuex applications](http://vuex.vuejs.org/en/structure.html" \t "http://vuejs-templates.github.io/webpack/_blank).

### **static/**

This directory is an escape hatch for static assets that you do not want to process with Webpack. They will be directly copied into the same directory where webpack-built assets are generated.

See [Handling Static Assets](http://vuejs-templates.github.io/webpack/static.html) for more details.

### **test/unit**

Contains unit test related files. See [Unit Testing](http://vuejs-templates.github.io/webpack/unit.html) for more details.

### **test/e2e**

Contains e2e test related files. See [End-to-end Testing](http://vuejs-templates.github.io/webpack/e2e.html) for more details.

### **index.html**

This is the ****template**** index.html for our single page application. During development and builds, Webpack will generate assets, and the URLs for those generated assets will be automatically injected into this template to render the final HTML.

### **package.json**

The NPM package meta file that contains all the build dependencies and [build commands](http://vuejs-templates.github.io/webpack/commands.html).