目录

[process.env.NODE\_ENV = 'production' 1](#_Toc2036)

[var config = require('../config') 1](#_Toc13437)

["build": "node build/build.js", 1](#_Toc17751)

[build | index.js envArr 2](#_Toc22104)

[module.exports.build 与 module.exports.dev 3](#_Toc979)

[var webpackConfig = require('./webpack.build.conf')](#_Toc6237) [4](#_Toc6237)

[exports.styleLoaders 5](#_Toc2177)

[Vue Loader 6](#_Toc24763)

[Introduction 6](#_Toc2125)

[VERSION NOTE 6](#_Toc16962)

[Migrating from v14 6](#_Toc32273)

[Heads Up注意 6](#_Toc23178)

[Notable Breaking Changes 6](#_Toc28323)

[#A Plugin is Now Required 7](#_Toc27200)

[#Loader Inference 7](#_Toc29277)

[SFC spec 单文件组件规范 https://vue-loader.vuejs.org/spec.html Vue Single-File Component (SFC) Spec 8](#_Toc16662)

[#Intro 8](#_Toc22506)

打包脉络

process.env.NODE\_ENV = 'production'

var config = require('../config')

"build": "node build/build.js",

require('./check-versions')()  
  
process.env.NODE\_ENV = 'production'  
  
var ora = require('ora')  
var rm = require('rimraf')  
var path = require('path')  
var chalk = require('chalk')  
var webpack = require('webpack')  
var config = require('../config')  
var webpackConfig = require('./webpack.build.conf')  
  
var spinner = ora('building for production...')  
spinner.start()  
  
rm(config.build.assetsRoot, err => {  
 if (err) throw err  
 webpack(webpackConfig, function (err, stats) {  
 spinner.stop()  
 if (err) throw err  
 process.stdout.write(stats.toString({  
 colors: true,  
 modules: false,  
 children: false,  
 chunks: false,  
 chunkModules: false  
 }) + '\n\n')  
  
 console.log(chalk.cyan(' Build complete.\n'))  
 console.log(chalk.yellow(  
 ' Tip: built files are meant to be served over an HTTP server.\n' +  
 ' Opening index.html over file:// won\'t work.\n'  
 ))  
 })  
})

build | index.js envArr

var envArr = {  
 production: require("./production.env"),  
 development: require("./development.env"),  
 stage: require("./stage.env"),  
 release: require("./release.env"),  
 testing: require("./testing.env")  
}

module.exports.build 与 module.exports.dev

module.exports = {  
 build: {  
 env: envArr[param],  
 index: path.resolve(\_\_dirname, '../dist/index.html'),  
 srcPath: path.resolve(\_\_dirname, '../src'),  
 assetsRoot: path.resolve(\_\_dirname, '../dist'),  
 assetsSubDirectory: 'static',  
 assetsPublicPath: '/',  
 productionSourceMap: false,  
 productionHtmlMinify: false,  
 // Gzip off by default as many popular static hosts such as  
 // Surge or Netlify already gzip all static assets for you.  
 // Before setting to `true`, make sure to:  
 // npm install --save-dev compression-webpack-plugin  
 productionGzip: false,  
 productionGzipExtensions: ['js', 'css'],  
 // Run the build command with an extra argument to  
 // View the bundle analyzer report after build finishes:  
 // `npm run build --report`  
 // Set to `true` or `false` to always turn it on or off  
 bundleAnalyzerReport: process.env.npm\_config\_report  
 },  
 dev: {  
 env: require('./development.env'),  
 port: 5500,  
 autoOpenBrowser: true,  
 assetsSubDirectory: 'static',  
 assetsPublicPath: '/',  
 proxyTable: {},  
 // CSS Sourcemaps off by default because relative paths are "buggy"  
 // with this option, according to the CSS-Loader README  
 // (https://github.com/webpack/css-loader#sourcemaps)  
 // In our experience, they generally work as expected,  
 // just be aware of this issue when enabling this option.  
 cssSourceMap: false  
 }  
}

var webpackConfig = require('./webpack.build.conf')

exports.styleLoaders

return {  
 css: generateLoaders(),  
 postcss: generateLoaders(),  
 less: generateLoaders('less'),  
 sass: generateLoaders('sass', { indentedSyntax: true }),  
 scss: generateLoaders('sass'),  
 stylus: generateLoaders('stylus'),  
 styl: generateLoaders('stylus')  
}

**[Vue Loader](https://vue-loader.vuejs.org/)**

Introduction

**VERSION NOTE**

This is the documentation for Vue Loader v15 and above. If you are upgrading from v14 or an earlier version, check out the [Migration Guide](https://vue-loader.vuejs.org/migrating.html). If you are using an older version, the old docs are [here](https://vue-loader-v14.vuejs.org/" \t "https://vue-loader.vuejs.org/_blank).

<https://vue-loader.vuejs.org/migrating.html>

# **Migrating from v14**

**Heads Up注意**

We are in the process of upgrading Vue CLI 3 beta to use webpack 4 + Vue Loader v15, so you might want to wait if you are planning to upgrade to Vue CLI

## **Notable Breaking Changes**

### [#](https://vue-loader.vuejs.org/migrating.html" \l "a-plugin-is-now-required)**A Plugin is Now Required**

Vue Loader v15 now requires an accompanying webpack plugin to function properly:

// webpack.config.jsconst { VueLoaderPlugin } = require('vue-loader')  
  
module.exports = {  
 // ...  
 plugins: [  
 new VueLoaderPlugin()  
 ]

}

### [#](https://vue-loader.vuejs.org/migrating.html" \l "a-plugin-is-now-required)**Loader Inference**

Vue Loader v15 now uses a different strategy to infer loaders to use for language blocks.

Take <style lang="less"> as an example: in v14 and below, it will attempt to load the block with less-loader, and implicitly chains css-loader and vue-style-loader after it, all using inline loader strings.

In v15, <style lang="less"> will behave as if it's an actual \*.less file being loaded. So, in order to process it, you need to provide an explicit rule in your main webpack config:

{  
 module: {  
 rules: [  
 // ... other rules  
 {  
 test: /\.less$/,  
 use: [  
 'vue-style-loader',  
 'css-loader',  
 'less-loader'  
 ]  
 }  
 ]  
 }}

The benefit is that this same rule also applies to plain \*.less imports from JavaScript, and you can configure options for these loaders anyway you want. In v14 and below, if you want to provide custom options to an inferred loader, you'd have to duplicate it under Vue Loader's own loaders option. In v15 it is no longer necessary.

v15 also allows using non-serializable options for loaders, which was not possible in previous versions.

# SFC spec [单文件组件规范](https://vue-loader.vuejs.org/zh/spec.html) https://vue-loader.vuejs.org/spec.htmlVue Single-File Component (SFC) Spec

## [#](https://vue-loader.vuejs.org/spec.html" \l "intro)**Intro**

A \*.vue file is a custom file format that uses HTML-like syntax to describe a Vue component. Each \*.vue file consists of three types of top-level language blocks: <template>, <script>, and <style>, and optionally additional custom blocks:

<template>  
 <div class="example">{{ msg }}</div>  
</template>  
  
<script>  
export default {  
 data () {  
 return {  
 msg: 'Hello world!'  
 }  
 }  
}  
</script>  
  
<style>  
.example {  
 color: red;  
}  
</style>  
  
<custom1>  
This could be e.g. documentation for the component.  
</custom1>

vue-loader will parse the file, extract each language block, pipe them through other loaders if necessary, and finally assemble them back into an ES Module whose default export is a Vue.js component options object.

vue-loader supports using non-default languages, such as CSS pre-processors (预处理器) and compile-to-HTML template languages, by specifying(设置) the lang attribute for a language block. For example, you can use Sass for the style of your component like this:

<style lang="sass">  
*/\* write Sass! \*/*</style>

More details can be found in [Using Pre-Processors](https://vue-loader.vuejs.org/guide/pre-processors.html).