



构建基础篇 2: webpack 在 CLI 3 中的应用

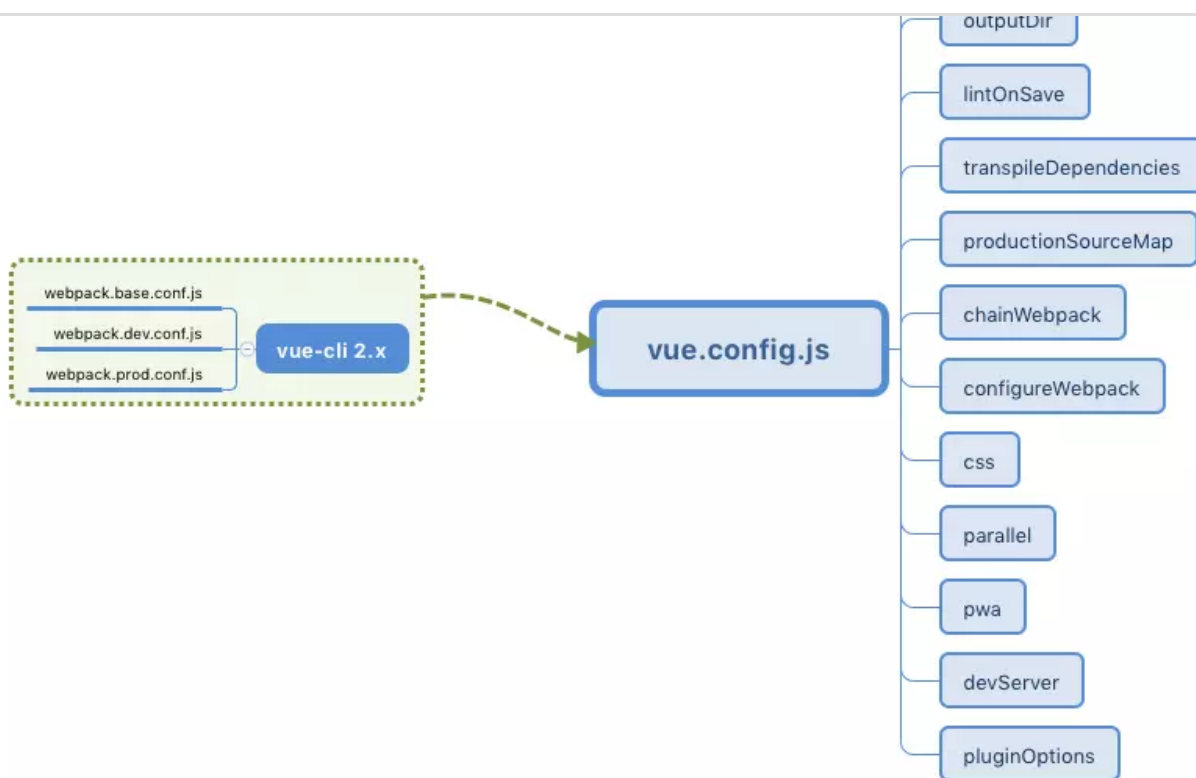
webpack 作为目前最流行的项目打包工具，被广泛使用于项目的构建和开发过程中，其实说它是打包工具有点大材小用了，我个人认为它是一个集前端自动化、模块化、组件化于一体的可拓展系统，你可以根据自己的需要进行一系列的配置和安装，最终实现你需要的功能并进行打包输出。

而在 Vue 的项目中，webpack 同样充当着举足轻重的作用，比如打包压缩、异步加载、模块化管理等等。如果你了解 webpack 那么相信本文会让你更了解其在 Vue 中的使用，如果你是一个 webpack 小白，那么也没事，相信你会很容易的了解它在项目中的配置和功能。

webpack 的使用

1. 与 vue-cli 2.x 的差异

如果你使用过 vue-cli 2.x，那么你应该了解其构建出的目录会包含相应的 webpack 配置文件，但是在 vue-cli 3.x 中你却见不到一份关于 webpack 的配置文件，难道 3.x 抛弃了 webpack？其实不然，3.x 提供了一种开箱即用的模式，即你无需配置 webpack 就可以运行项目，并且它提供了一个 vue.config.js 文件来满足开发者对其封装的 webpack 默认配置的修改。如图：



2. vue.config.js 的配置

通过上方新老版本的对比，我们可以清晰的看出 `vue.config.js` 的配置项结构，如果你构建的项目中没有该文件，那么你需要在根目录手动创建它。下面我们就来介绍一下其常用配置项的功能和用途：

a. baseUrl

在第一节《Vue CLI 3 项目构建基础》中我们通过 `vue-cli 3.x` 成功构建并在浏览器中打开 <http://localhost:8080/> 展示了项目首页。如果现在你想要将项目地址加一个二级目录，比如：<http://localhost:8080/vue/>，那么我们需要在 `vue.config.js` 里配置 `baseUrl` 这一项：

```
// vue.config.js
module.exports = {
  ...

  baseUrl: 'vue',
```

javascript



其改变的其实是 webpack 配置文件中 output 的 `publicPath` 项，这时候你重启终端再次打开页面的时候我们首页的 url 就会变成带二级目录的形式。

b. outputDir

如果你想将构建好的文件打包输出到 output 文件夹下（默认是 dist 文件夹），你可以配置：

```
                                javascript
// vue.config.js
module.exports = {
  ...

  outputDir: 'output',

  ...
}
```

然后运行命令 `yarn build` 进行打包输出，你会发现项目跟目录会创建 output 文件夹，这其实改变了 webpack 配置中 output 下的 `path` 项，修改了文件的输出路径。

c. productionSourceMap

该配置项用于设置是否为生产环境构建生成 source map，一般在生产环境下为了快速定位错误信息，我们都会开启 source map：

```
                                javascript
// vue.config.js
module.exports = {
  ...

  productionSourceMap: true,

  ...
}
```

该配置会修改 webpack 中 `devtool` 项的值为 `source-map`。

d. chainWebpack

chainWebpack 配置项允许我们更细粒度的控制 webpack 的内部配置，其集成的是 [webpack-chain](#) 这一插件，该插件可以让我们能够使用链式操作来修改配置，比如：

```
                                javascript
// 用于做相应的合并处理
const merge = require('webpack-merge');

module.exports = {
  ...

  // config 参数为已经解析好的 webpack 配置
  chainWebpack: config => {
    config.module
      .rule('images')
      .use('url-loader')
      .tap(options =>
        merge(options, {
          limit: 5120,
        })
      )
  }

  ...
}
```

以上操作我们可以成功修改 webpack 中 module 项里配置 rules 规则为图片下的 url-loader 值，将其 limit 限制改为 5M，修改后的 webpack 配置代码如下：

```
                                javascript
{
  ...

  module: {
    rules: [
      {
        /* config.module.rule('images') */
        test: /\. (png|jpe?g|gif|webp) (\?.*)?$/,
        use: [
```

```
        options: {
          limit: 5120,
          name: 'img/[name].[hash:8].[ext]'
```

这里需要注意的是我们使用了 `webpack-merge` 这一插件，该插件用于做 `webpack` 配置的合并处理，这样 `options` 下面的其他值就不会被覆盖或改变。

关于 `webpack-chain` 的使用可以参考其 github 官方地址：[github.com/mozilla-neu...](https://github.com/mozilla-neutrons/webpack-chain)，它提供了操作类似 JavaScript `Set` 和 `Map` 的方式，以及一系列速记方法。



configureWebpack 更倾向于整体替换和修改。示例代码如下：

javascript

```
// vue.config.js
module.exports = {
  ...

  // config 参数为已经解析好的 webpack 配置
  configureWebpack: config => {
    // config.plugins = []; // 这样会直接将 plugins 置空

    // 使用 return 一个对象会通过 webpack-merge 进行合并, plugins 不会置空
    return {
      plugins: []
    }
  }

  ...
}
```

configureWebpack 可以直接是一个对象，也可以是一个函数，如果是对象它会直接使用 webpack-merge 对其进行合并处理，如果是函数，你可以直接使用其 config 参数来修改 webpack 中的配置，或者返回一个对象来进行 merge 处理。

你可以在项目目录下运行 `vue inspect` 来查看你修改后的 webpack 完整配置，当然你也可以缩小审查范围，比如：

shell

```
# 只查看 plugins 的内容
vue inspect plugins
```

f. devServer

vue.config.js 还提供了 devServer 项用于配置 webpack-dev-server 的行为，使得我们可以对本地服务器进行相应配置，我们在命令行中运行的 `yarn serve` 对应的命令 `vue-cli-service serve` 其实便是基于 webpack-dev-server 开启的一个本地服务器，其常用配置参数如下：



```
...

devServer: {
  open: true, // 是否自动打开浏览器页面
  host: '0.0.0.0', // 指定使用一个 host。默认是 localhost
  port: 8080, // 端口地址
  https: false, // 使用https提供服务
  proxy: null, // string | Object 代理设置

  // 提供在服务器内部的其他中间件之前执行自定义中间件的能力
  before: app => {
    // `app` 是一个 express 实例
  }
}

...
}
```

当然除了以上参数，其支持所有的 `webpack-dev-server` 中的选项，比如 `historyApiFallback` 用于重写路由（会在后续的多页应用配置中讲解）、`progress` 将运行进度输出到控制台等，具体可参考：[devServer](#)

以上讲解了 `vue.config.js` 中一些常用的配置项功能，具体的配置实现需要结合实际项目进行，完整的配置项可以查看：[vue.config.js](#)

3. 默认插件简介

通过对 `vue.config.js` 的了解，我们知道了 `vue-cli 3.x` 为我们默认封装了项目运行的常用 `webpack` 配置，那么它给我们提供了哪些默认插件，每一个 `plugin` 又有着怎样的用途呢？除了使用 `vue inspect plugins` 我们还可以通过运行 `vue ui` 进入可视化页面查看，步骤如下：

- 打开可视化页面，点击对应项目进入管理页面（如果没有对应项目，需要导入或新建）
- 点击侧边栏 `Tasks` 选项，再点击二级栏 `inspect` 选项
- 点击 `Run task` 按钮执行审查命令

如图所示：



最后我们从输出的内容中找到 `plugins` 数组，其包含了如下插件（配置项已经省略，增加了定义插件的代码）：

javascript

```
// vue-loader是 webpack 的加载器，允许你以单文件组件的格式编写 Vue 组件
const VueLoaderPlugin = require('vue-loader/lib/plugin');

// webpack 内置插件，用于创建在编译时可以配置的全局常量
const { DefinePlugin } = require('webpack');

// 用于强制所有模块的完整路径必需与磁盘上实际路径的确切大小写相匹配
const CaseSensitivePathsPlugin = require('case-sensitive-paths-webpack-plugin');

// 识别某些类型的 webpack 错误并整理，以提供开发人员更好的体验。
const FriendlyErrorsPlugin = require('friendly-errors-webpack-plugin');

// 将 CSS 提取到单独的文件中，为每个包含 CSS 的 JS 文件创建一个 CSS 文件
const MiniCssExtractPlugin = require("mini-css-extract-plugin");

// 用于在 webpack 构建期间优化、最小化 CSS文件
const OptimizeCssnanoPlugin = require('optimize-css-assets-webpack-plugin');

// webpack 内置插件，用于根据模块的相对路径生成 hash 作为模块 id，一般用于生产环境
const { HashedModuleIdsPlugin } = require('webpack');

// 用于根据模板或使用加载器生成 HTML 文件
const HtmlWebpackPlugin = require('html-webpack-plugin');

// 用于在使用 html-webpack-plugin 生成的 html 中添加 <link rel='preload'> 或 <link rel='prefetch'>
const PreloadPlugin = require('preload-webpack-plugin');

// 用于将单个文件或整个目录复制到构建目录
```




```
plugins: [  
  /* config.plugin('vue-loader') */  
  new VueLoaderPlugin(),  
  
  /* config.plugin('define') */  
  new DefinePlugin(),  
  
  /* config.plugin('case-sensitive-paths') */  
  new CaseSensitivePathsPlugin(),  
  
  /* config.plugin('friendly-errors') */  
  new FriendlyErrorsWebpackPlugin(),  
  
  /* config.plugin('extract-css') */  
  new MiniCssExtractPlugin(),  
  
  /* config.plugin('optimize-css') */  
  new OptimizeCssnanoPlugin(),  
  
  /* config.plugin('hash-module-ids') */  
  new HashedModuleIdsPlugin(),  
  
  /* config.plugin('html') */  
  new HtmlWebpackPlugin(),  
  
  /* config.plugin('preload') */  
  new PreloadPlugin(),  
  
  /* config.plugin('copy') */  
  new CopyWebpackPlugin()  
]  
}
```

我们可以看到每个插件上方都添加了使用 `chainWebpack` 访问的方式，同时我也添加了每个插件相应的用途注释，需要注意的是要区分 `webpack` 内置插件和第三方插件的区别，如果是内置插件则无需安装下载，而外部插件大家可以直接访问：www.npmjs.com/ 搜索对应的插件，了解其详细的 `api` 设置。



展后续内容的学习，为接下来项目的构建和开发奠定基础。

思考 & 作业

- 除了文章中介绍的配置项，`vue.config.js` 中还有哪些额外的配置？
- `webpack-merge` 的合并原理是怎样的？
- 使用 `chainWebpack` 获取到 webpack 中的某一插件后，如何修改其配置？

留言

评论将在后台进行审核，审核通过后对所有人可见

chenlinfei .NET攻城狮 @ 深圳某互联网金融科技...

通过vue-cli3新建的项目，怎么找不到vue.config.js 文件在哪？

▲ 0 收起评论 21天前

oscar-liu

需要自己新增一个，参考vue-cli官方文档

20天前

大绵羊 码农 @ 科技公司

大哥，文章里说的很明白啊，要自己在项目目录下创建一个啊

10小时前

评论审核通过后显示

评论



▲ 0 评论 29天前

萧雪圣 前端开发工程师

关于 baseUrl 的配置，我在vue.config.js 里配置后重新运行项目 提示

“baseUrl” option in vue.config.js is deprecated now, please use “publicPath” instead.
好像已经弃用了

▲ 1 收起评论 1月前

FrangFan

3.3以后官方推荐用publicPath <https://cli.vuejs.org/config/#baseurl>

1月前

评论审核通过后显示

评论

努力的彭小猿 前端开发

思考&作业是不是应该给出答案

▲ 0 评论 1月前

Ricci web前端开发

截止到这一章节，我运行npm run build 之后报错，编译后依赖的css 和 JS 路径为啥不对呢 Refused to apply style from 'http://localhost:63343/css/app.9640e964.css' because its MIME type ('text/html') is not a supported stylesheet MIME type, and strict MIME checking is enabled.

index.html:20 GET http://localhost:63343/js/app.29d3fe3d.js net::ERR_ABORTED 404 (... [展开全部](#))

▲ 0 评论 2月前

hongxb

limit: 5120 “将其 limit 限制改为 5M” 你是认真的吗？



劳卜 前端工程师 @ TC

这里仅以此举例

2月前

评论审核通过后显示

评论

李雨冰 前端 @ MYCOMPANY

运行了vue ui命令后，如何把已经创建好的项目导入，在导入界面没有可操作的地方，请指教。

▲ 0 收起评论 2月前

劳卜 前端工程师 @ TC

顶部tab切换到“导入”，有导入按钮

2月前

评论审核通过后显示

评论

啊Mark 前端开发

为什么我搭建的cli3没有vue.config.js文件

▲ 0 收起评论 2月前

WeitongWong

自行创建一个

2月前

评论审核通过后显示

评论

为这世界操碎了心 前端

想请问一下 image-webpack-loader 怎么配置啊



劳卜 前端工程师 @ TC

可以参考 webpack-chain 创建 loader 部分：

<https://github.com/neutrinojs/webpack-chain#config-module-rules-uses-loaders-creating>

2月前

评论审核通过后显示

评论

gaspar本尊

文中边线是曲线的文本框好看，请问是用啥工具画的？

▲ 0

收起评论 3月前

劳卜 前端工程师 @ TC

wireframeSketcher

3月前

评论审核通过后显示

评论

为这世界操碎了心 前端

想请问一下，我的生产环境资源引用是cdn地址，但是我配置了baseUrl后，本地开发的资源引用也变了，该怎么解决？

▲ 0

收起评论 3月前

劳卜 前端工程师 @ TC

可以区分环境配置 baseUrl

3月前

为这世界操碎了心 前端

好的，谢谢

3月前

benhh 前端



为这世界操碎了心 前端

嗯嗯，我直接这样写了 `baseUrl:process.env.VUE_APP_CDN`

3月前

评论审核通过后显示

评论

Si

// 提供在服务器内部的其他中间件之前执行自定义中间件的能力

```
before: app => {  
  // `app` 是一个 express 实例  
}
```

你好,请问这句话是什么意思,能举个具体的例子吗?

▲ 0 收起评论 3月前

劳卜 前端工程师 @ TC

可以参考: <https://webpack.docschina.org/configuration/dev-server/#devserver-before>

3月前

评论审核通过后显示

评论

雪宝宝的外卖丢了 前端工程师菜鸡

小白一个，我运行yarn build，生成的是dist文件夹，跟 output 文件夹是一样的吧？

▲ 1 收起评论 3月前

劳卜 前端工程师 @ TC

嗯，不配置的话默认是 dist

3月前



xoyoz 前端工程师

作业的答案在哪可以看到？

▲ 0 收起评论 3月前

劳卜 前端工程师 @ TC

没有固定的答案地址，主要用于读者思考和评论区讨论的

3月前

评论审核通过后显示

评论

agenting1473742110000 htmlllllllllll @ 飞云

关于这个vue.config.js的完整代码有没有完整实例，比如d. chainWebpack下面修改代码不明白，复制进去无法运行

▲ 0 收起评论 3月前

劳卜 前端工程师 @ TC

可以参考小册的实例代码：<https://github.com/luozhihao/vue-project-code/tree/master/single-page-project>

3月前

守一座空城 前端开发 @ 软通动力

为什么我vue-cli后没有vue.config.js文件？这河文件要自己手动创建的吗

3月前

劳卜 前端工程师 @ TC

回复 守一座空城：根据你初始化时选的配置，可能会没有，没有就需要手动创建

3月前

评论审核通过后显示

评论



劳卜 前端工程师 @ TC

和官网表述不同的，做了整理和拓展

3月前

鹰之重生

webpack配置还不如官方文档全面

3月前

评论审核通过后显示

评论

llc1994 前端菜狗子 @ 图元科技

完整的vue.config.js那个地址是404 --

▲ 0

收起评论 3月前

劳卜 前端工程师 @ TC

已经修改了

3月前

评论审核通过后显示

评论

Linmi 导游 @ 掘金

不知道怎么加入《Vue 项目构建与开发入门》作者群的小伙伴可以看这里：

<https://ws1.sinaimg.cn/large/006tNbRwly1fxn1g79tmqj30u00xdh0l.jpg>

▲ 0

收起评论 3月前

Linmi 导游 @ 掘金

已购买的用户才会出现加群按钮哈

3月前

评论审核通过后显示

评论

