### 使用webpack产生的疑问

###### 现在的项目为什么和我之前使用的webpack.base.config不一样了？

调整 webpack 配置最简单的方式就是在 vue.config.js 中的 configureWebpack 选项提供一个对象：

configureWebpack: config => {}

Vue CLI 内部的 webpack 配置是通过 webpack-chain 维护的。这个库提供了一个 webpack 原始配置的上层抽象，使其可以定义具名的 loader 规则和具名插件，并有机会在后期进入这些规则并对它们的选项进行修改。它允许我们更细粒度的控制其内部配置。

chainWebpack: config => {}

vuecli3使用的是webpack4

###### 我想配置使用一个babel-plugin-lodash为什么不起作用？

Vuecli3中使用了bable7中的心配置格式 babel.config.js。

和.bablel和package.json中的bable不同，这个配置文件不会使用基于文件位置的方案，而是会一致的运用到项目根目录下的所有文件，包括node\_modules内部依赖，所有推荐在vueCli项目中始终使用babel.config.js取代其他格式

babel.config.js

module.exports = {

     presets: [ "@vue/app" ],

     Plugins: [‘lodash’]

};

package.json

{

...

"devDependencies": {

"babel-plugin-lodash": "^3.3.4",

}

...

}

###### 在vueCli3中想使用HappyPack搜索发现不需要了

在vueCli3中的vue.config.js中内部存在一个parallel默认开机多线程构建(生成环境)