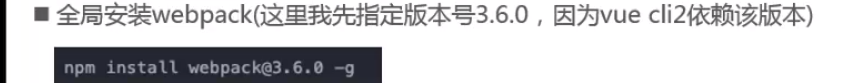
1. 入门介绍

Webpack：前端模块化自动管理工具，相比于grunt,gulp更强大，grunt,gulp更倾向于任务式的去进行压缩之类的处理。而这在webpack属于附加功能，不过相比之下它也更大一点。





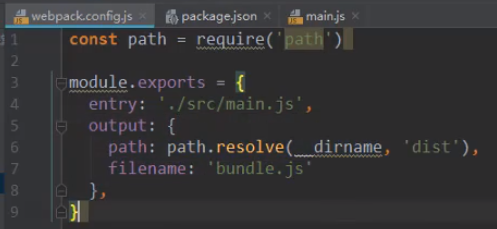
使用时需要两个文件夹src和dist，dist为发布文件夹放打包后的js，src为开发时的文件，另需个index..html直接引用dist里的js即可。

1. webpack.config.js配置和package.json配置

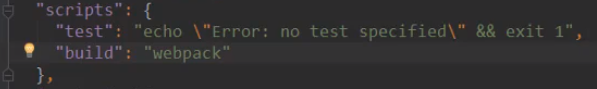
配置该文件的主要目的就是为了更方便的进行打包。

先配置webpack.config.js文件，再不进行映射的情况下，这个文件名是固定的。

先在终端输入npm init进行初始化生成package.json文件，如果package.json中有什么依赖的话要再输入npm install来安装node\_modules。

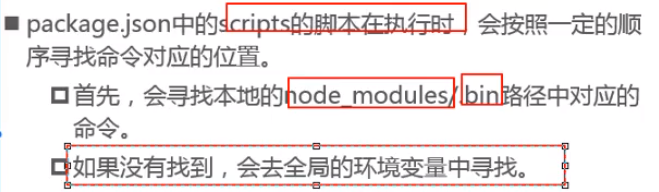


这里除了entry和output中的路径，其它都为固定写法。这样之后可以直接敲webpack进行打包。不过还有更简单的写法就是用npm写法去进行映射。具体操作为在package.json的script属性中去增加快捷命令以及对应的webpack命令。例如下图，这样执行npm run bulid其实就相当于执行webpack。



**下载局部webpack并进行调用**

在当前路径终端下输入npm install webpack@版本号 --save-dev（这是开发时依赖），这时如果直接输webpack始终调用的是全局的webpack，想要使用局部的要用npm run bulid这种脚本写法



1. webpack中使用css文件的配置

首先需要用npm下载loader，具体的有npm install -–save-dev style-loader/css-loader，之后在入口文件中添加require并在config.js文件添加module对象





4、webpack对于sass，less文件的处理

在入口文件添加依赖的less文件，npm下载less相关的loader再在config.js的rules中加入相应的规则即可。

5、webpack对于图片文件的处理

先用npm下载url-loader与file-loader，再在config文件的rules中进行相关配置



这里的限制的字节数，当图片大小小于他时就将图片编译成base64进行展示，大于它就选择用文件引用。，publicPath是在config.js文件的出口的参数中加。

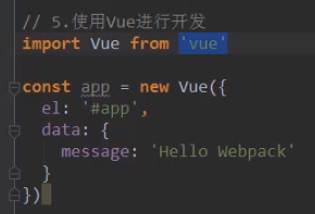
6、webpackES6转ES5的babel



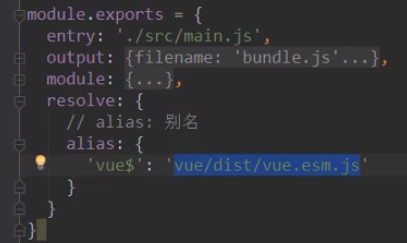
exclude(排除的意思)

7、Webpack配置Vue环境

先npm install vue –save安装vue，然后在入口文件导入vue

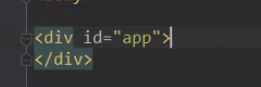


这样直接运行会报错，因为默认使用的vue版本不会编译初始模板需要在config.js文件中重新指定

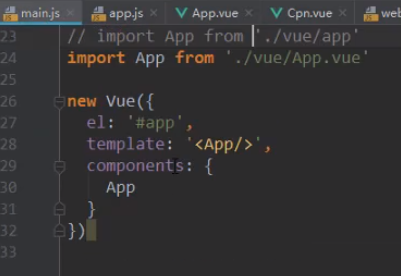


8、Vue的终极使用方案

创建多个vue文件作为组件并构成组件树，一般根组件起名叫App.vue，组件中可以构建模板，样式，以及导出的js代码，当然也可以在script标签中引入别的组件，总体上就是组件树的开发思想，最后入口文件导入App.vue文件中的导出对象，并进行引用即可。这样index.html文件仅仅需要定义一个id为app的div与入口文件的vue实例形成挂钩即可。



下图的对象中新增了template属性，当该属性存在时，会用该模板去替换原来的id为app的元素块。



当然，新增了vue文件所以需要通过npm下载新的loader并配置config.js文件，compiler是编译的意思





9、plugin（插件的使用）

插件有的是webpack自带，有的则需要通过npm安装，对于自带的直接通过require获得node\_modules中的内容即可，不自带的则要先npm安装再require。

（1）版权插件



（2）index.html插件

这里的模板路径是相对于webpack.config.js文件的路径来定的



（3）压缩js插件

10、webpack-dev-server搭建本地服务器

这个本地服务器的作用主要是避免每次更改都需要输入npm run bulid，通过服务器可以在更改后实现实时刷新。



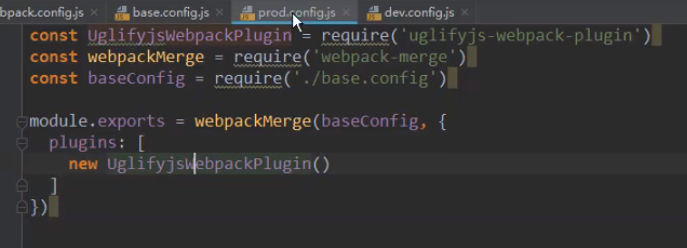
这里又添加了个快捷dev命令，因为默认的webpack-dev-server命令是从全局找。

11、webpack配置文件的分离

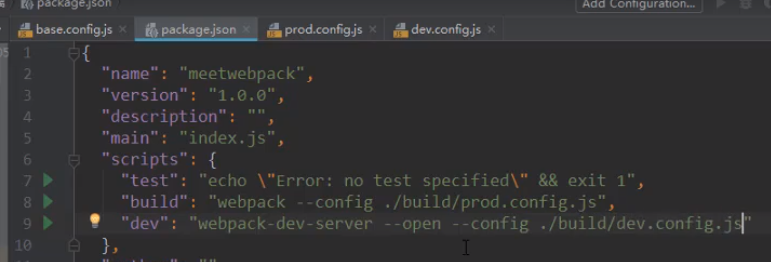
即将开发时需要的文件分为一部分，再将发布时需要的文件再分一部分

可以新建三个js文件，分别命名为base.config.js，dev.config.js，pro.config.js，将开发环境与生产环境中的公共东西放在base文件中，之后根据情况将开发中的与生产中的分别放入dev与pro中。

要先通过npm安装合并模块，npm install webpack-merge –save-dev，之后在dev与pro文件中分别进行下图中的类似合并



这样之后删除了webpack-config.js，因此需要去package.json中去重新指定路径



注意现在要注意出口文件的路径，因为出口路径是相当于配置文件的路径决定的

12、Vue CLI脚手架的介绍与安装

脚手架需要node与webpack环境





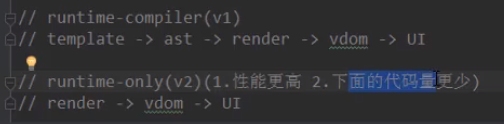


13、VueCLI2



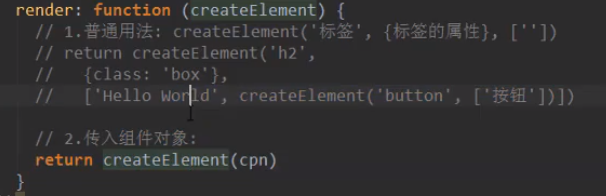
第四步的Vue bulid，暂时建议选带有runtime compiler的那个。不过真正的公司开发一般就选runtime only。

**（1）runtime only 与runtime compiler的区别**

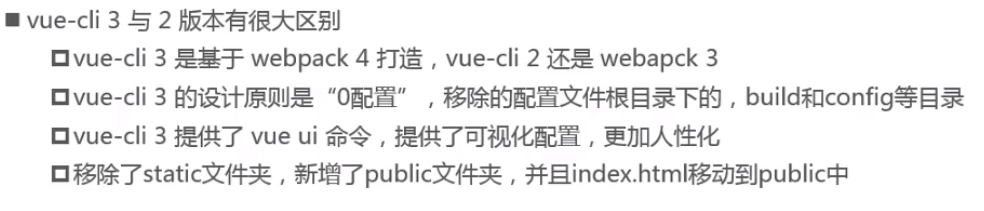
二者过程对比

render函数详解

render:function (h){ return h(App)}，这里的h其实是简写，它的原有名称为createElement，下图为用法，这里当直接传入组件对象时会创建组件对象。

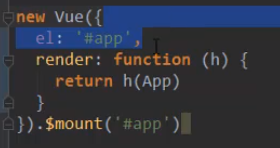


14、VueCLI3创建项目和目录结构





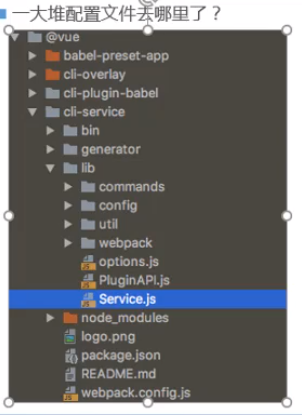
这里一般选手动配因为不想要这个eslint。

这里的el和$mount(“#app”)选择任何一个都可以，el的本质就是在内部将其转化为mount这种形式，所以或许mount更快。

**（1）VueCLI3配置文件的查看与修改**

一，命令行输入vue ui，需要先导入文件夹才能进行相关查看和修改；

二，node\_modules里可以查看；



三，自定义配置文件，新建vue.config.js，然后配置好文件进行导出

