一.安装webpack的两种方式

1>.npm install webpack –g(全局安装)

2>.npm install webpack –save-dev(安装到项目依赖中)

二．webpack的基本使用

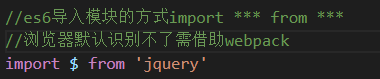
在终端使用命令(以jqeury项目为例)

1>. npm init –y(初始化项目)

2>.npm install jquery –S(将jqeury包安装到生产环境依赖)

(S, save的缩写意思为安装到dependencies(生产阶段的依赖))

3>.在导入文件中写导入文件🡪



注意：引入css文件不需要from后面的东西见6>.

4>.webpack 要打包的文件路径 打包到哪里(引用这个文件) (打包之后能使用es6的语法)

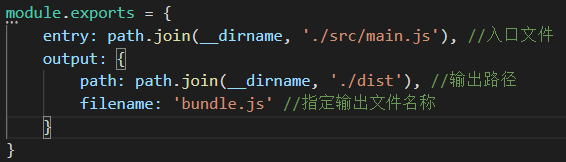
可配置自动打包(每次都要重新打包) 配置自动打包安装webpack-dev-server工具可实现-🡪

1.创建webpack.config.js配置文件(根目录下)

2.cnpm install webpack-dev-server –D(安装时候会出现警告 需要本地安装webpack依赖)

3.安装cnpm install webpack –D

3.在webpack.config.js文件设置入口和出口文件-🡪



配置完成后可以使用webpack命令直接打包 不需要指定打包文件和打包到哪里见4>.

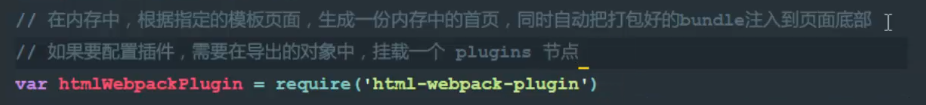
4.设置自动打包🡪

在package.json文件中添加一个脚本

“dev”: “webpack-dev-server --open(自动打开浏览器) --port 3000(设置端口) – contentBase src(默认打开哪src文件) –-hot(热更新)(设置其可以不刷新页面更新css样式) ”

5>.cnpm install html-webpack-plugin –D(托管第三方工具托管在内存中)—>

webpack.config.js中配置导入



在model.exports对象中与entry同级创建plugins对象-🡪



6>.cnpm install style-loader css-loader 安装第三方loader

(直接在main.js入口函数导入css文件会报错，需安装第三方loader)

配置loader -🡪和entry同级



三．webpack中url-loader的使用(不能处理css文件中的路径样式)

cnpm install url-loader file-loader (安装依赖) 配置🡪

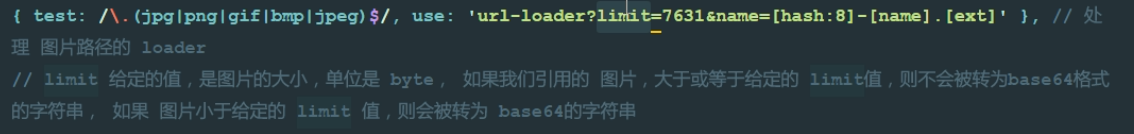


添加参数

1>.第一个参数(图片名是网络地址)



2>.第二个参数(图片名是自己设置图片的名字)



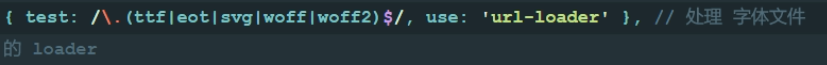
四.安装bootstrp使用它的字体图标

cnpm install bootstrap –S (导入bootstrap)

1>.引用

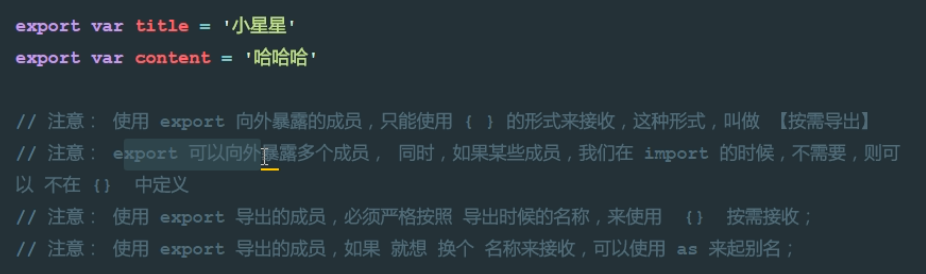


配置🡪



五．export和export default暴露成员的区别

Export🡪



Export default🡪



引用 Expotr需有{} 而Export default 不需要{}🡪

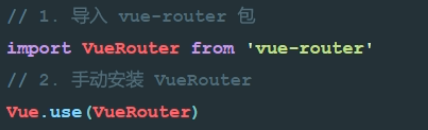




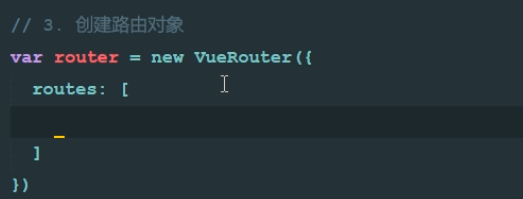
六.webpack使用路由

1>.安装路由 npm install vue-router –S

2>.导入router包，且手动安装路由包(在入口js main.js)



3>.创建路由对象🡪(在入口js main.js)



注意：(rander函数的特性原因)

