webpack会自动处理模块之间的依赖

首先现在nodejs,配置环境变量。Nodejs自带npm包，用其安装webpack

命令：

npm install webpack@版本 -g //全局安装

npm install webpack@版本 --save-dev //局部安装

为什么全局安装后还需要局部安装？

1. 在终端直接执行webpack命令，使用的是全局安装的webpack
2. 当在package.json中定义了scripts时，其中包含了webpack命令，使用的是局部安装的webpack

webpack项目目录

src（源文件目录） 开发目录

main.js或者index.js（入口）

dist(打包文件夹)

index.html

打包命令

webpack ./src/main.js ./dist/bundle.js

js文件导出导入语法

1. commonjs模块化规范语法

module.exports={

}

const {\*} = require(“\*.js”)

1. ES6语法

export const name=’’;

export const age=’’;

import {name,age} from “\*.js”;

webpack配置文件

文件名 webpack.config.js

导入path模块

import const path = require(‘path’)

命令 npm init 运行之后会生成package.json

{

  "name": "meetwebpack",

  "version": "1.0.0",

  "description": "",

  "main": "index.js",

  "scripts": {

    "test": "echo \"Error: no test specified\" && exit 1"

  },

  "author": "",

  "license": "ISC"

}

scripts就是设置执行脚本的,可以设置各种脚本

npm run test;

里面的脚本都是优先执行本地的webpack

有可能本地和全局的版本号不一致,比如自己电脑的版本是3.6 但是下载下来的项目是3.7,就会有冲突

安装本地webpack

命令:npm install --save-dev [webpack@3.6.0](mailto:webpack@3.6.0) 如果安装不成功,把node\_modules文件夹删除

卸载本地 npm un webpack

 const path = require("path") //commonjs语法

//\_\_dirname是当前文件的路径

//resolve()拼接

module.exports={

    entry:'./src/main.js',

    output:{

        path:path.resolve(\_\_dirname,"dist"),

        filename:'bundle.js'

    }

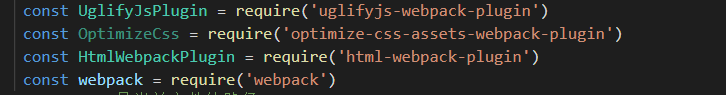
}

插件

1. 安装对应的插件,注意版本冲突

npn install 插件名 --save-dev

1. 导入插件



1. 创建插件对象

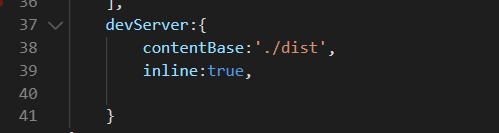


本地服务

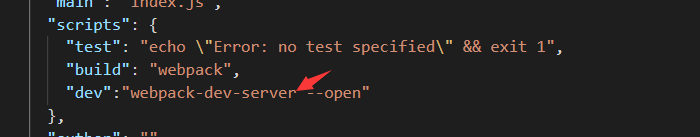
1. 引入webpack模块

const webpack = require('webpack')

1. 创建本地服务



1. 修改配置文件



启动

npm run dev

退出服务

ctrl+C

vue脚手架

1. 安装脚手架

npm install -g @vue/cli 最新版

npm install -g @vue/cli-init 拉去2.x的版本

1. 初始化项目 使用2

vue init webpack 项目名

 回车

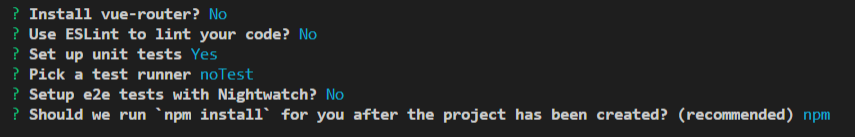
随意





随意

 N



vue路由