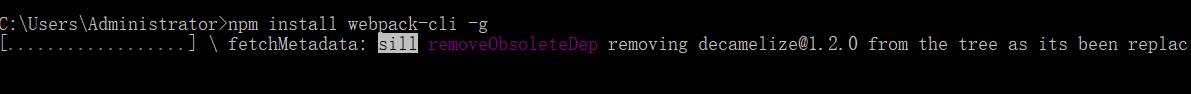
webpack

1 安装与基本命令

* 1. 安装命令

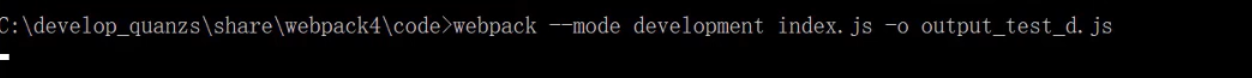
npm install webpack-cli -g



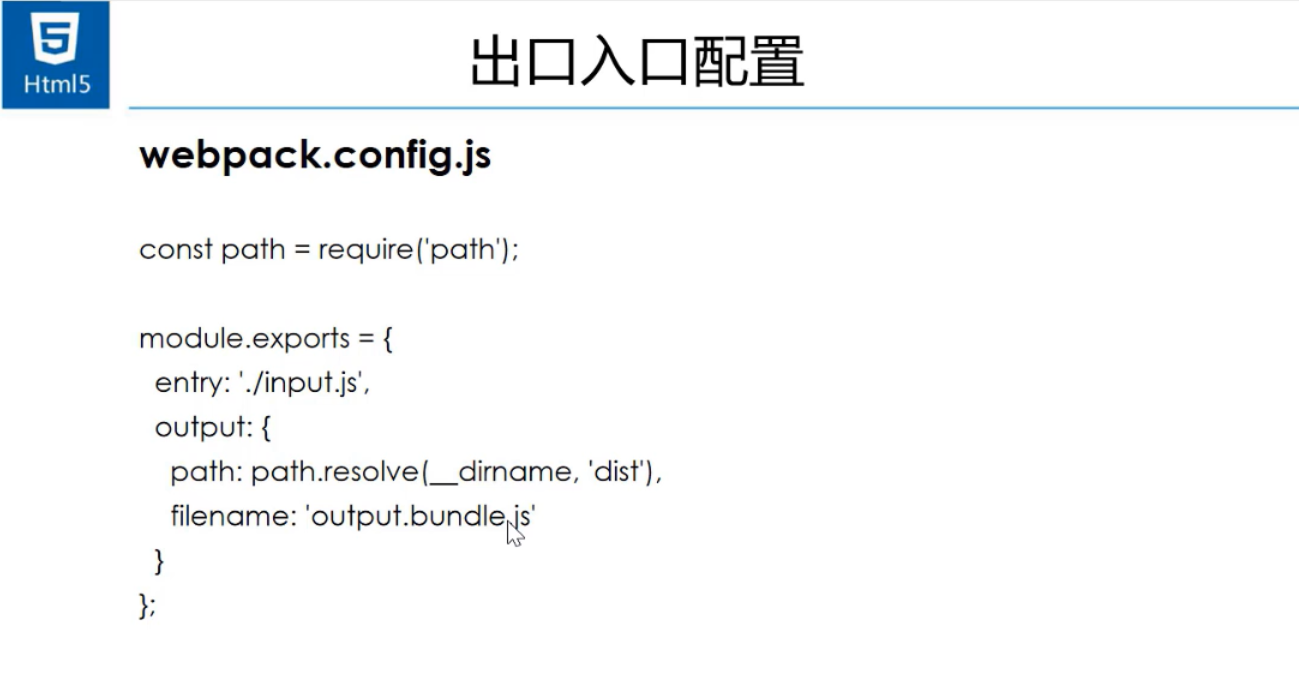
慢 可以使用 淘宝镜像

基本使用

webpack输入文件.js 输出.js



1. 配置出入口文件
   1. webpack.config.js



*const* path = require('path');

module\_exports={

entry:path.join(\_\_dirname,'./src/main.js'),

output:{

path: path.join(\_\_dirname,'dist'),

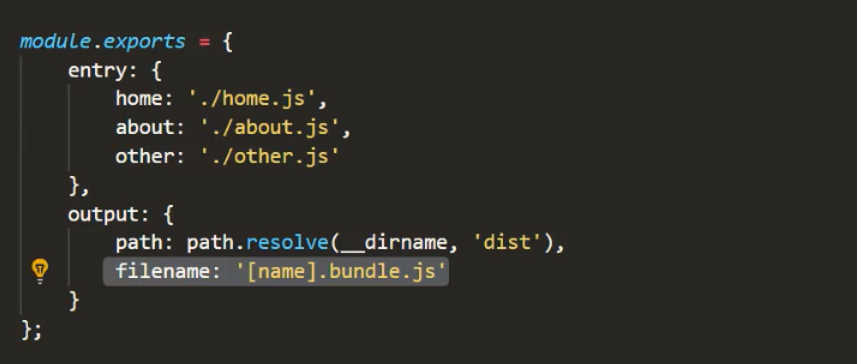
filename:'bundle.js'

}

};

以上失败

多个文件打包



ERROR in Entry module not found: Error: Can't resolve './src' in 'D:\Study\webpackstudy'

报错

解决: module.export 而不是 module\_export

*const* path = require('path');

*module*.*exports*={

entry:{

home:'./src/home.js',

about:'./src/about.js',

other:'./src/other.js'

},

output:{

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

filename:'[name].bundle.js'

},

mode: "development"

};

The CLI moved into a separate package: webpack-cli.

Please install 'webpack-cli' in addition to webpack itself to use the CLI.

-> When using npm: npm install webpack-cli -D

-> When using yarn: yarn add webpack-cli -D

首先下载来的webpack版本是4.6.0  
我写好package.json

"scripts": {

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

"dev": "webpack-dev-server --open", // npm run dev 热更新

"build": "webpack -p" // npm run build 打包

},

执行语句npm run dev，发现有错误，啊，我才开始学，就报错，内容如下

要命啊，我不懂啊，我之前命名全局安装过脚手架，为什么他现在又要我安装一次，哎？-D是什么鬼，好的，百度  
-D原来就是--save-dev。。。。  
百度后，得知，webpack4.0开始就有些不同，根据网上经验者提及，如

在webpack 3中，webpack本身和CLI都在同一个包中，但是在版本4中，他们将两者分开来更好地管理它们。

所以我要重新装一次CLI....

在项目目录下要本地安装webpack  
npm install webpack-cli -D

webpack.config.js文件的变化  
webpack.config.js里的配置，不再支持 module下的loaders，需要把loaders改成rules。  
在项目目录下必须使用配置文件webpack.config.js

即如下：

// webpack 必须采用 commonjs 写法

//专门处理路径用的，以当前路径解析出一个绝对路径

let path = require('path');

console.log(path.resolve('./dist'));

console.log(path.resolve('./dist'));

module.exports = {

//打包的入口文件webpack会自动查找相关的依赖进行打包

entry: './src/main.js',

output: {

filename: 'bundle.js', //打包后的名字

path: path.resolve('./dist'), //必须使用个绝对路径

},

module: {

rules: [

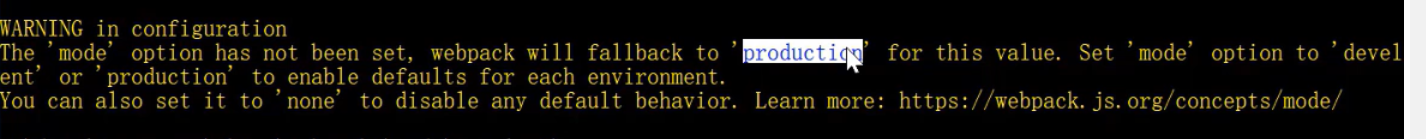
//针对css文件，进行对应的loader处理

{ test: /\.css$/, loader: "style-loader!css-loader" }

]

}

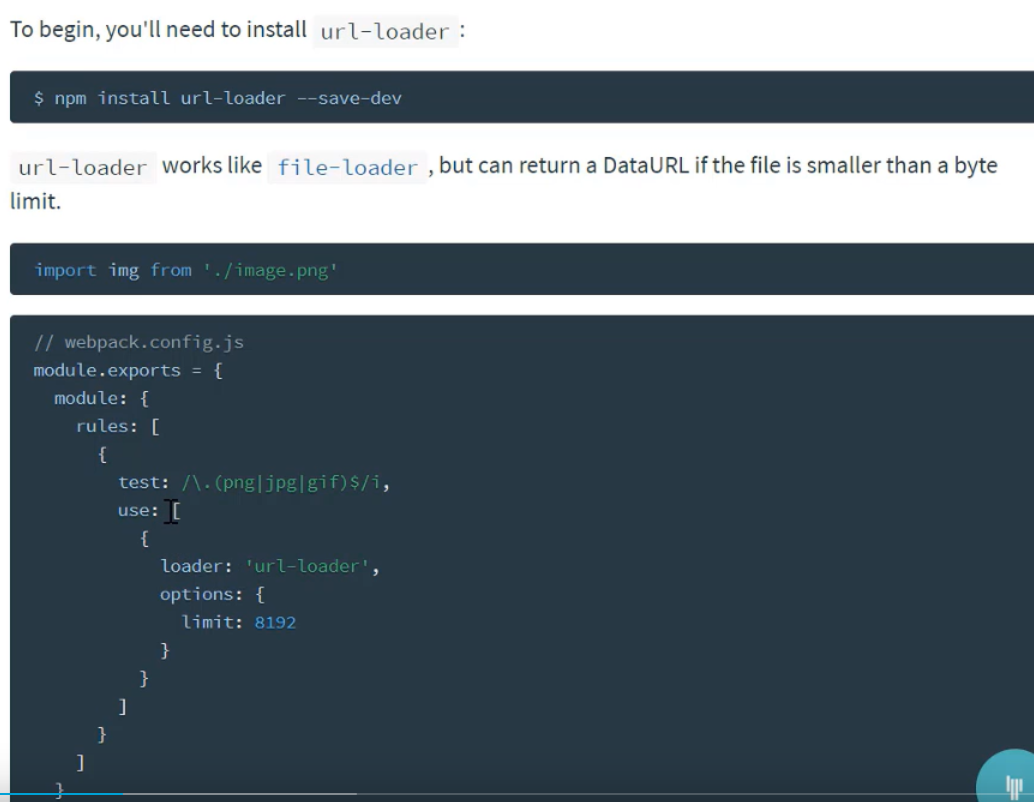
};



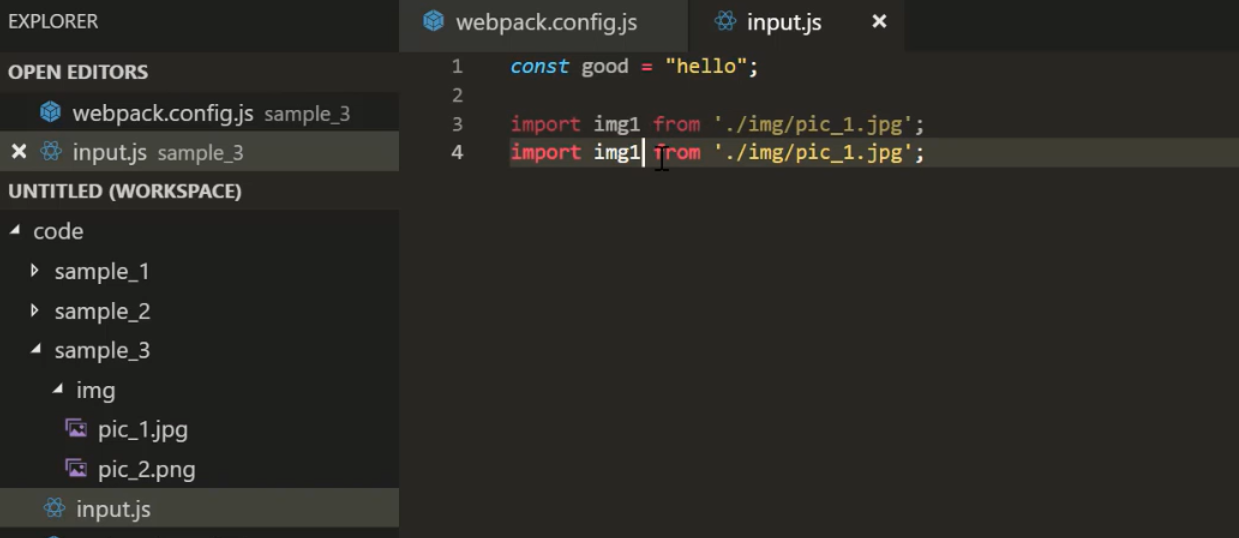
webpack – mode development 开发代码 有助于发现问题



1. 加载器 Loaders
   1. url-loader



1 在入口文件entry (.js) 中引用图片



1. npm init -y 这样会生成一个文件 package.json
2. 配置webpack规则

eg: **Wrote to D:\Study\webpackstudy\package.json:**

**{**

**"name": "webpackstudy",**

**"version": "1.0.0",**

**"main": "index.js",**

**"scripts": {**

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

**"dev": "webpack-dev-server --open",**

**"build": "webpack -p"**

**},**

**"keywords": [],**

**"author": "",**

**"license": "ISC",**

**"dependencies": {**

**"jquery": "^3.4.1"**

**},**

**"devDependencies": {**

**"webpack": "^4.35.0",**

**"webpack-cli": "^3.3.5"**

**},**

**"description": ""**

**}**

**在webpack.config.js中配置**

module:{

rules:[

{

test:/\.(png)|(jpg)$/i,

use:[

{

loader:'url-loader',

options: {

limit:8192

}

}

]

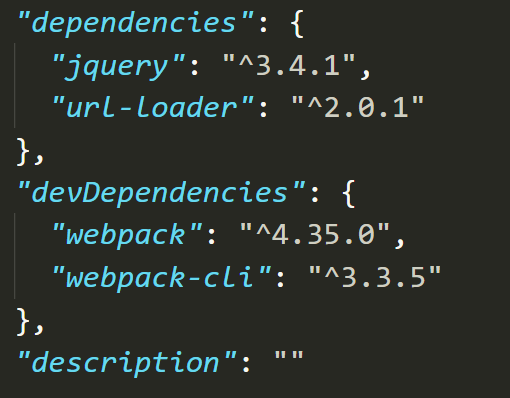
}

]

4 安装url-loader

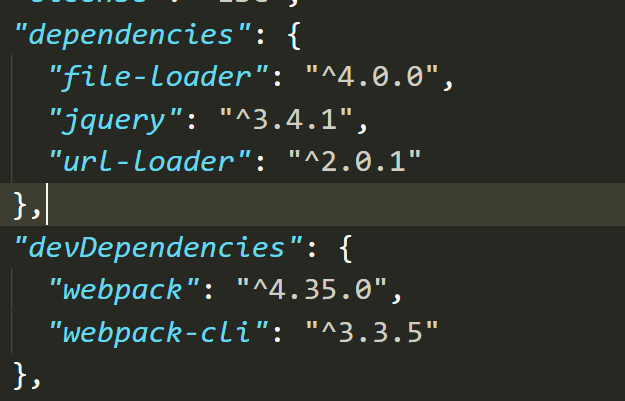
npm install url-loader -S

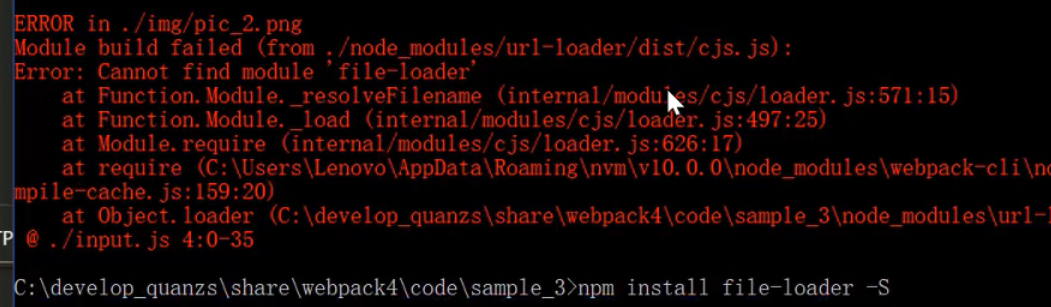
纪录到文件中去=>package.json中



webpack执行之后再 进行

npm install file-loader -S





在执行一次webpack

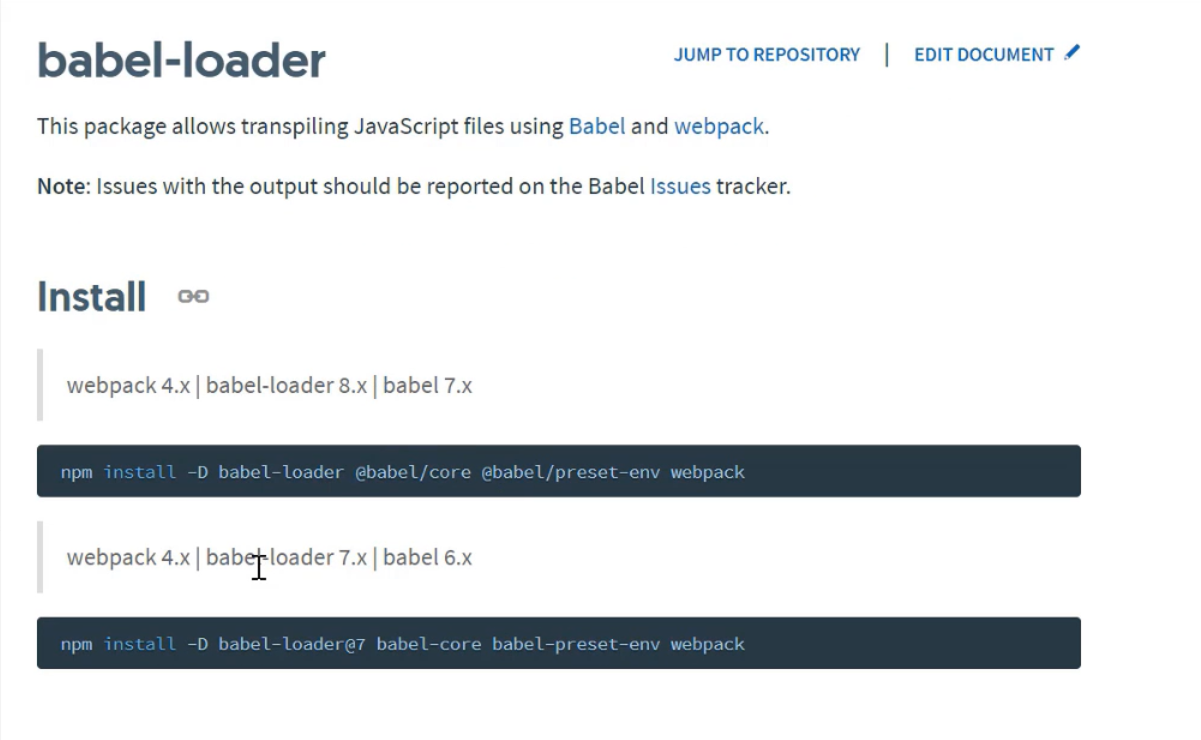
设置了一个limit 小于limit 限制的大小 => base64 大于limit就变成普通的目录文件



* 1. bable loader

3.2.1 基本命令

实际就是把ES6语法转换成普通的浏览器能解析 的代码

推荐最新的 (7.x) 版本

npm install -D babel-loader @babel/core @babel/preset-env webpack

3.2.3 使用



preset :规则

在input.js 新写一个规则



async *function* sayHello(){

*const* result = await fetch("http://www.baidu.com");

*console*.log(result)

await 做异步处理

