webpack笔记

一.webpack安装

npm install 或 yarn add webpack webpack-cli –D(本地安装)

二.webpack的功能

解析js中的模块，并且以当前js为准，查找相关依赖的文件，打包成一个文件，并且解决了浏览器require的问题。

三.webpack配置

1.webpack的0配置打包很弱,并且只支持js模块

2.手动配置webpack

--默认配置文件的名字，webpack.config.js

--//webpack用的是node的语法

第一步:配置入口文件(entry)

第二步:配置出口(olutput)文件,并配置打包模式(mode) production是生产模式 development是开发模式

3.webpack-dev-serve配置本地的服务器

第一步

安装webpack-dev-server

命令行: yarn add/npm install webapck-dev-server –D

作用:生成一种内存中的打包

第二步

在webpack.config.js中配置devServer对象(称:开发服务的配置)

在package.json的scripts配置启动插件



第三步

安装 html-webpack-plugin插件,命令行 yarn add/npm install html-webpack-plugin –D

并引入html-webpack-plugin模块

第四步

在plugin数组里面配置相关插件



3.webpack样式文件配置

loder的作用:把原代码转化成一个模块，特点是希望单一，如果使用单一loader就使用字符串,例子use:’css-loader’,如果使用多个loader使用数组use:[‘css-loader’,’style-loader’]

注意：loader的执行顺序是右向左执行,从下到上执行

第一步：

安装loader:命令行 yarn add /npm install css-loader style-loader less-loader –D

第二步:在module中配置css style less scss 等loader



备注:css-loader处理css中特殊语法例如@import，style-loader把解析后的css插入到style标签中

第三步：

配置抽离css的插件，作用是预防阻塞

1.安装

命令行：yarn add/npm install mini-css-extract-plugin –D

2.引入依赖，并在plugin内设置相关属性,通过mini-css-extract-plugin.loader属性取代style-loader

第四步:

1.给样式属性添加前缀。

命令行：yarn add postcss-loader autoprefixer –D

2.在项目中新建postcss.config.js

3.在postcss.config.js中引入插件模块 autoprefixer

4.CSS压缩

第一步：

安装css,js的压缩模块

命令行：

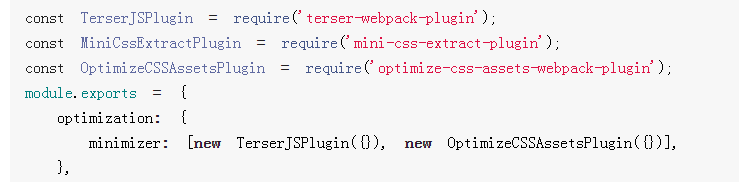
yarn add/npm install optimize-css-assets-webpack-plugin –D

yarn add/npm install terser-webpack-plugin –D

并在webpack.config.js引入两个模块

第二步：

在optimization的minimizer属性中引入



5.JS处理

1.安装babel-loader(进行转化的加载器),@babel/core(bable的核心模块),@babel/preset-env(把高级的语法转化成低级语法)

命令行

yarn add/npm install babel-loader @babel/core @babel/preset-env –D

yarn add/npm install @babel/plugin-proposal-class-properties –D

(es7=>es5)

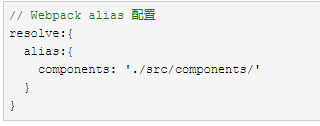
2.在module中配置相关设置

四.webpack 中 Resolve配置

作用:配置Webpack如何寻找模块对应的文件。

属性:

1.resolve.alias:配置项通过别名把原导入路径映射成一个新的导入路径。



以上例子，当通过import button from ‘components/button’会被解析成

import button from ‘./src/components/button’

2.resolve.mainFields

作用:让webpack根据mainFields的配置去决定优先采用哪一份代码。

首先要在package.json里面设置



语法:mainFields:[‘browser’,’main’]（默认）

如果要先采用es6的语法的代码配置如下:

mainFields:[‘jsnext:main’, ‘browser’,’main’]