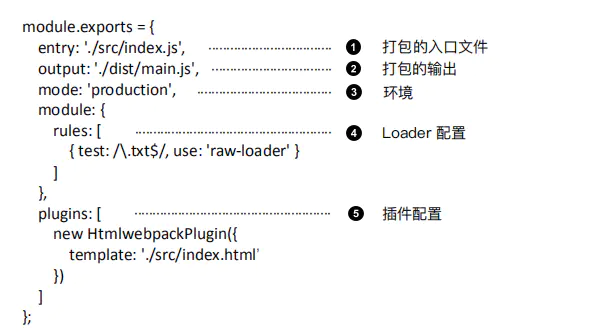
webpack

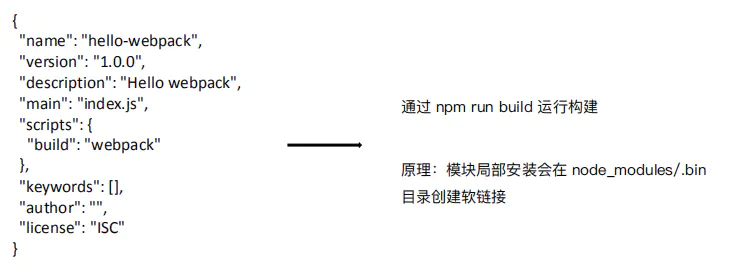
2021年6月26日

20:35

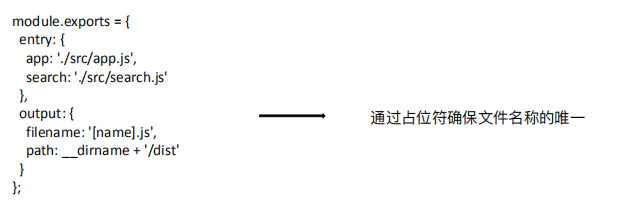
* + **webpack配置构成**



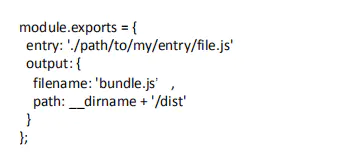
* + **如何打包?**
    - 使用./node\_modules/.bin/webpack运行打包
    - 配置package.json配置



* + **entry与output配置**



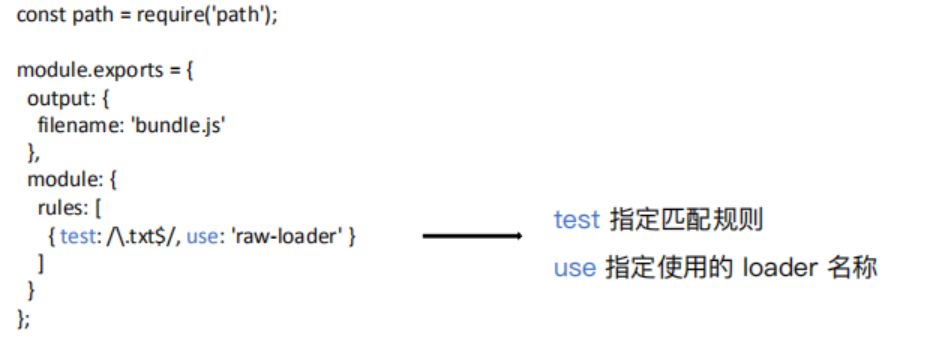
**或者**



* + **loader**
    - **常用loader**



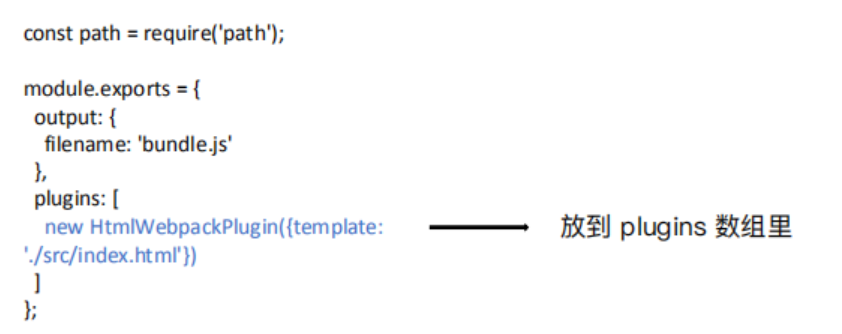
* + **loader配置**



* + **plugin**
    - **常用plugin**



* + **plugin用法**



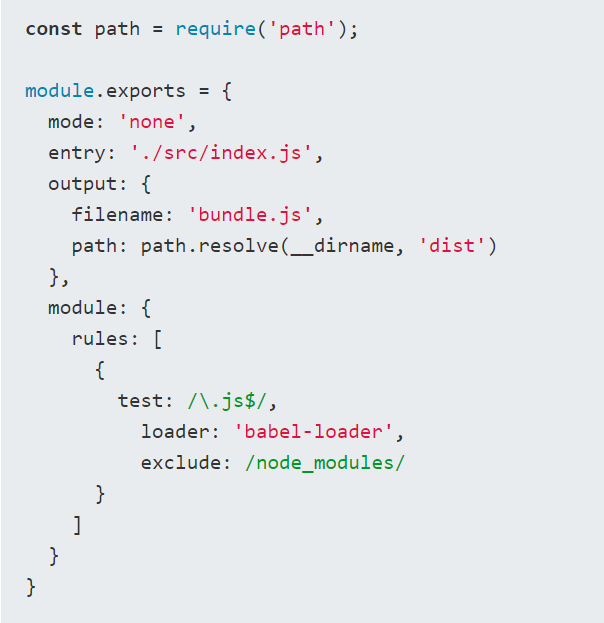
* + **Mode**



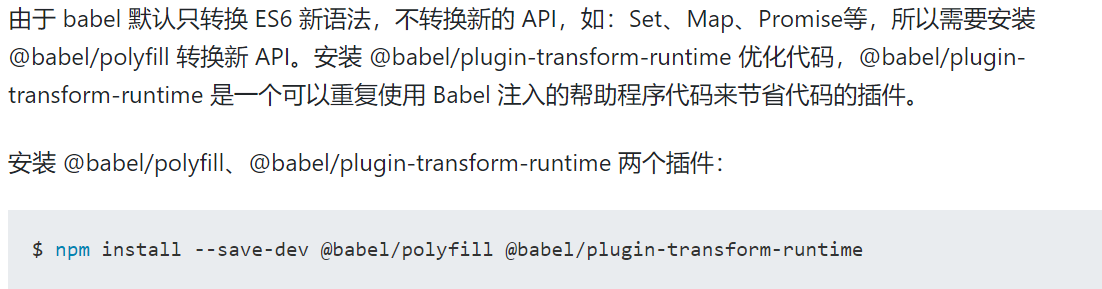
* + **解析ES6**
    - **安装@babel/core、@babel/preset-env、babel-loader**

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

* + **webpack配置**



* + **.babelrc配置**



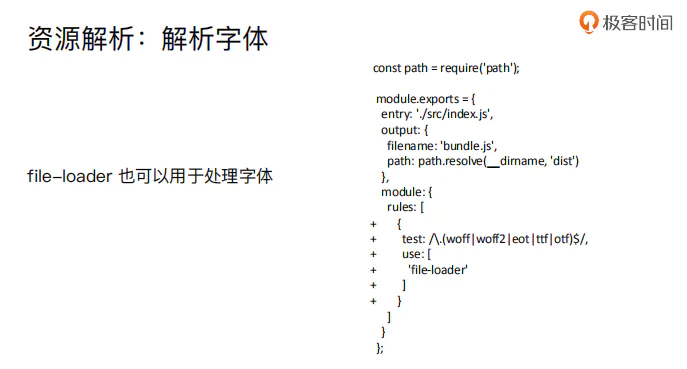
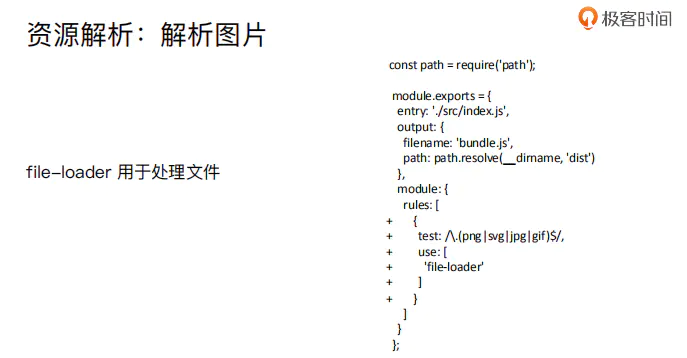
* + **解析CSS**
    - **解析css样式要安装style-loader、css-loader**
    - **webpack配置**



* + **解析less与sass**
    - **已解析less为例，安装less、less-loader**
    - **webpack配置**



* + **解析图片和字体**
    - **安装file-loader**
    - **webpack配置**



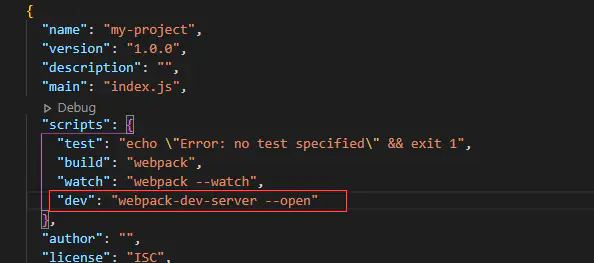
* + **或者使用url-loader**

图中的limit(字节)是设置图片的最小大小，如果图片的大小小于limit设置的值url-loader会将图片解析成base64在页面上进行展示。

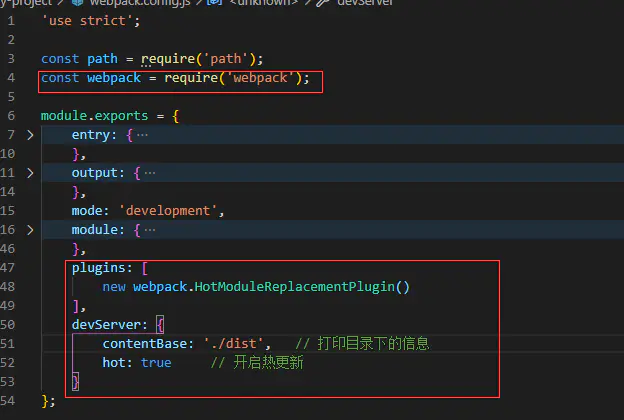


* + **webpack-dev-server**
    - 在package.json中添加一个dev命令，配置好webpack-dev-server;

npm i webpack-dev-server -D



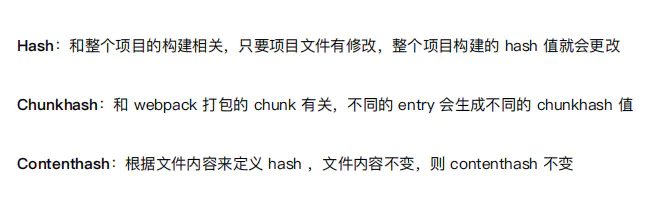
* + 修改webpack.config.js中的mode为development
  + 配置HotModuleReplacementPlugin插件和devServer(无需安装)



* + 运行npm run dev查看运行结果
  + 报错 Cannot find module ‘webpack-cli/bin/config-yargs ,则webpack-dev-server 改写成 webpack serve
  + **webpack-dev-middleware(热更新,通常适用express、koa等后端服务，需要引用nodejs)**



* + **文件指纹策略**
    - **指纹类型**



一般js用Chunkhash，css用Contenthash

* + **JS指纹**



* + **CSS文件指纹**

npm i mini-css-extract-plugin -D

如果正常的情况下我们使用style-loader、css-loader的话，那么这个css会由style-loader插入到html文件的head头部，但是并没有独立的css文件.

使用MiniCssExtractPlugin这个插件，将head头部里面的css样式提取出来生成一个css文件。



* + **图片的指纹**



* + **HTML,CSS,JS代码压缩**
    - **JS文件压缩**

使用内置的uglifyjs-webpack-plugin

* + **CSS文件压缩**

安装依赖

npm i optimize-css-assets-webpack-plugin cssnano -D



* + **HTML文件的压缩**

安装依赖

npm i html-webpack-plugin -D

