Webpack静态模块打包器

——详见xlytest项目的webpack.config.js

学习记录：<https://www.webpackjs.com/guides/caching/>

1. 简单属性介绍
   1. entry[入口文件]
      1. 单个入口语法：entry:String/Array<String>
         1. main：入口

const config = {

entry: {

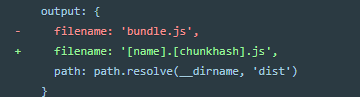
main: './path/to/my/entry/file.js'

}};

* + 1. 对象入口语法：entry:{[entryChunkName:String]:String|ArrayString}  
        这是应用程序中定义入口的最可扩展的方式
       1. app.js应用程序和vendor.js第三方库入口
  1. 缓存

Webpack的缓存：项目打包之后会生成./dist的文件夹，当./dist目录部署到服务器上的时候，在客户端/浏览器中就能够访问此网站及其资源，由于获取数据（前台请求后台接口）是项目里最费时的，所以基本上现在的浏览器都会带缓存，降低网络流量，等待下一次访问的时候会调用缓存数据，加速网站的加载速度，但如果项目重新打包部署了，原来的缓存数据还在，要获取新的数据就不好获取了，而通过配置webpack编译生成的文件能够被客户端缓存，而在文件内容变化后，能够请求到新的文件。

* 1. 输出文件的文件名

output:{filename:’文件名’}  


1. 插件
   1. clean-webpack-plugin:清除文件  
       下载：npm install --save-dev clean-webpack-plugin  
       引入：const CleanWebpackPlugin = require('clean-webpack-plugin')  
       plugin模块引入：new CleanWebpackPlugin(['build'])  
       运行打包命令：webpack --mode development
   2. html-webpack-plugin:创建文件入口  
      <https://www.jianshu.com/p/08a60756ffda>  
       为html文件中引入的外部资源如script、link动态添加每次compile后的hash，防止引用缓存的外部文件问题；可以生成创建html入口文件，比如单页面可以生成一个html文件入口，配置N个html-webpack-plugin可以生成N个页面入口
      1. 属性
         1. title:标题  
            就是我们写html的<head><title>标题<title></head>
         2. filename:html的文件名
         3. templete:模板(模板可以任意选择，需要安装对应的loader)  
            <https://www.webpackjs.com/concepts/loaders/>
         4. Inject:注入选项【这个不是特别懂，大概是选择html的js插入位置】
            1. true:默认值script标签位于html文件的 body 底部
            2. body:同true
            3. head:script放置head标签内
            4. false:不插入js文件
         5. favicon:html页面地址栏的图标  
            <https://blog.csdn.net/woshixuye/article/details/9218107>