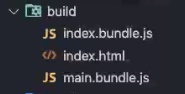


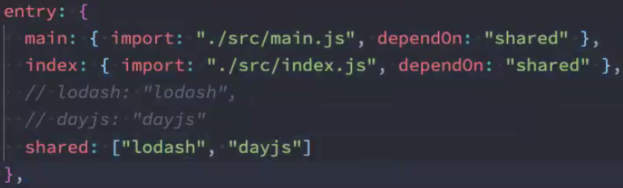
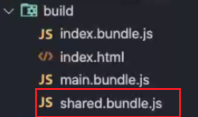
**entry方式**

**Entry Dependencies方式**

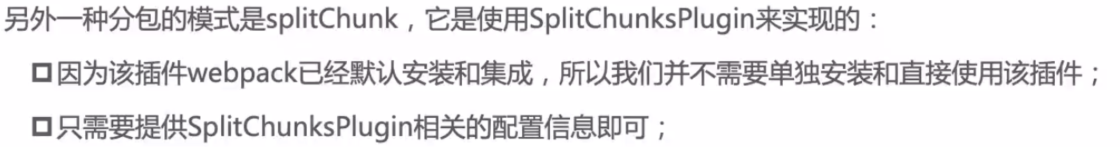
*当index.js和main.js中都引入一个库时，打包后两个文件中都会有库的源代码。*

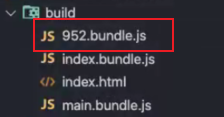
因此这样设置：



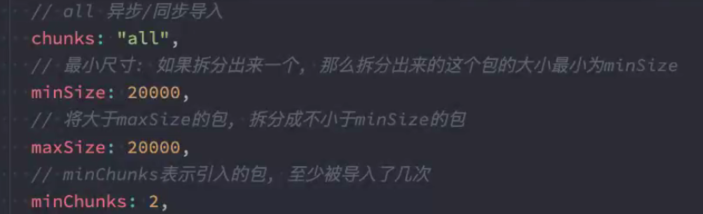
**SplitChunks**



Async时只会对公共的异步导入的代码进行分离  
all时公共 导入的所有代码会进行分离

**其他属性**

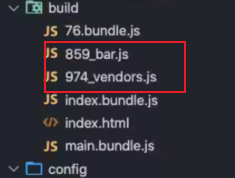
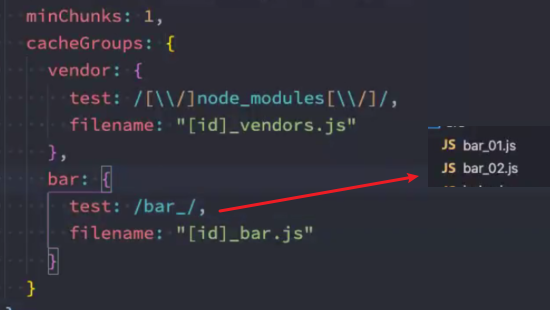


**minSize:**如果分离的代码大小小于20000字节就不会分离出来。

**maxSize**:如果分离的代码大小大于20000字节，就会拆分成不小于minsize的包。

**minChunks**:只有被导入2次及以上的包，才会被分离出来。

**cacheGroups:缓存组**



匹配到对应文件时就会将对应文件进行抽离

name是写死的，filename可以使用占位属性。



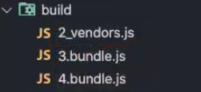
引入两次及以上的包分离到common中，如果上面和下面都满足，根据priority大小来进行优先级

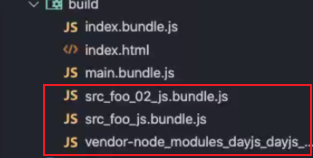
**异步导入的代码，webpack都会对代码进行分离**

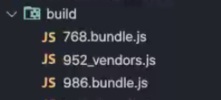


**通过chunkIds来控制分离代码打包的文件名**

**自然数**



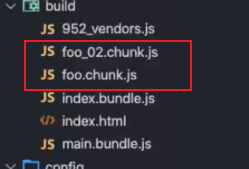
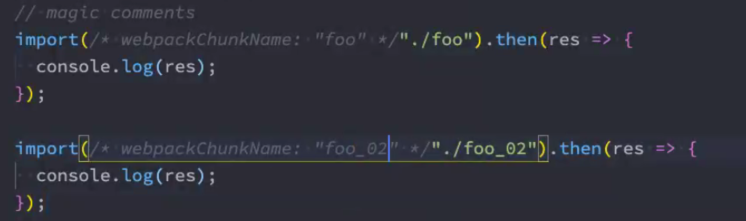




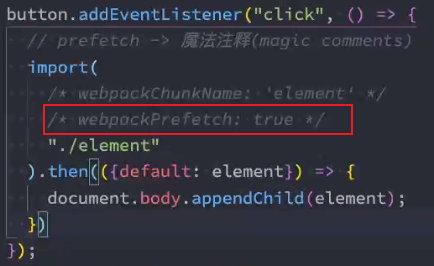
**异步加载导入的代码的打包后文件名通过chunkFilename设置**



**魔法注释修改名字**

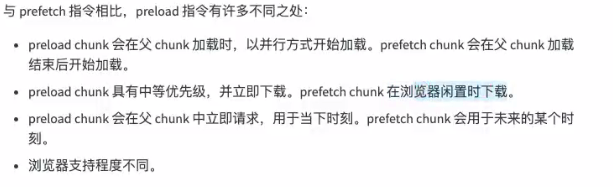


**懒加载**

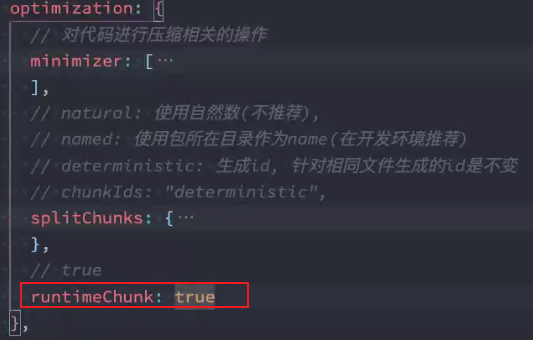


webpackPrefetch设为true后会在浏览器下载完其他文件空闲后，再下载element.js文件

webpackPreload与webpackPrefetch比较



optimization.runtimeChunk配置



true/multiple打包到多个包里面

single打包到一个包里面

对象：

