**node\_modules包不编译**

一般情况下，node\_modules安装的包都已经被打包为CommonJS规范的代码，代码的入口文件由包下package.json的main字段，如果你发现node\_modules安装的包不是ES5语法（如使用了class，使用了jsx），那么你完了，你将无法通过编译，因为babel7以上会默认不编译node\_modules目录下的代码（大多数情况下，我们的webpack的配置也是不打包node\_modules目录的），所以我们需要配置include字段，包含我们要打包的目录

              include: [

                paths.appSrc,

                path.resolve('node\_modules/react-native-webview/')

              ],

**不要在package.json中配置babel**

在package.json中配置babel让我遇到了一些奇葩的错误，还是使用官方的配置文件babel.config.js配置比较好，如下赠送一个示例：

module.exports = {

    presets: [

        "react-app",

        "module:metro-react-native-babel-preset"

    ],

    plugins: [

        [

            "module-resolver",

            {

                "root": [

                    "src/Common",

                    "src/RNApp",

                    "src/WebApp"

                ],

                "extensions": [

                    ".js",

                    ".jsx",

                    ".ts",

                    ".tsx"

                ]

            }

        ]

    ]

}

**alias先于Babel插件**

如果你在babel插件中配置了模块路径，你不能在alias应用该路径

let resolve = {

    alias: {

        // 将 react-native 的组件替换掉，以便 react-native 在 web 展示

        'react-native$': 'react-native-web',

        'react-router-native$': 'react-router-dom',

        'react-native-webview$': 'src/WebApp/IEWebView',

    },

},