**基本配置：**

module.exports = {

    entry: {

        account: \_\_dirname + "/Account/account.jsx",

        Account1: \_\_dirname + "/Account/account1.jsx"

    },

    output: {

        // 输出文件目录路径

        path: path.join(\_\_dirname, "js"),

        // 输出文件名

        filename: "[name].bundle.js",

        // 非入口文件名

        // 什么是非入口文件，如：

        // app.js 时入口文件，其在一个函数中使用了 import() 函数，动态加载（函数执行时加载）另一个 js 文件

        // 此时"另一个 js 文件"便是非入口文件

        chunkFilename: "[id].chunk.js"

    }

}

Output有很多字段，详细配置可查阅官网

## **清理 /build 文件夹**

如果你想在每次执行打包命令时都清了之前生成的文件，那么你可以使用如下插件

安装npm包

npm install clean-webpack-plugin --save-dev

**webpack.config.js**

const path = require('path');

+ const CleanWebpackPlugin = require('clean-webpack-plugin');

module.exports = {

entry: {

app: './src/index.js',

print: './src/print.js'

},

plugins: [

+ new CleanWebpackPlugin(['build'])

],

output: {

filename: '[name].bundle.js',

path: path.resolve(\_\_dirname, 'dist')

}

};

## **html页面的引入**

如果你想在打包命令时自动将打包的js文件的引用添加到html页面中，那么你可以使用如下方法

安装 html-webpack-plugin 插件

npm install html-webpack-plugin --save

安装 webpack-utf8-bom 插件

npm install webpack-utf8-bom --save

webpack.config.js配置

var path = require("path");

const HtmlWebpackPlugin = require('html-webpack-plugin') //HtmlWebpackPlugin插件

const BomPlugin = require('webpack-utf8-bom'); //将文件转成utf-8 bom格式，解决中文乱码的问题

module.exports = {

entry: {

account:\_\_dirname + "/Account/account.js"

},

output: {

path: \_\_dirname + "/build",

filename: "[name].bundle.js"

},

plugins: [

//这里开始写

new HtmlWebpackPlugin({

filename: \_\_dirname + '/build/login.html', //html生成路径,

template: \_\_dirname + '/Account/login.html', // html模板路径,

inject: 'body', //打包之后的js插入的位置，true/'head'/'body'/false,

chunks: ['account'] //需要引入的js

}),

new BomPlugin(true, /\.(cshtml)$/),//解决cshtml中文乱码的问题

......

]

......

}