# webpack.config.js 参数详解

## entry

entry可以是个字符串或数组或者是对象。

当entry是个字符串的时候，用来定义入口文件

entry: 'webpack/hot/dev-server',

当entry是个数组的时候，里面同样包含入口js文件，另外一个参数可以是用来配置webpack提供的一个静态资源服务器，webpack-dev-server。webpack-dev-server会监控项目中每一个文件的变化，实时的进行构建，并且自动刷新页面：

entry: ['webpack/hot/dev-server', path.resolve(\_\_dirname, './src/js/main.js')],

当entry是个对象的时候，我们可以将不同的文件构建成不同的文件，按需使用，比如在我的hello页面中只要\引入hello.js即可：

entry: {

hello: './js/hello.js',

form: './js/form.js'

}

## output

output: {  
 path: path.resolve(\_\_dirname, './build'), //编译路径  
 filename: 'bundle.js' //编译生成的js文件名称  
},

## module

关于模块的加载相关，我们就定义在module.rules中。这里通过正则表达式去匹配不同后缀的文件名，然后给它们定义不同的加载器。比如说给less文件定义串联的三个加载器（！用来定义级联关系）

module: {  
 rules: [  
 {  
 test: /\.js?$/,  
 exclude: /(node\_modules|bower\_components)/,  
 loader: 'babel-loader',  
 }  
 ],  
},

## 4、resolve

webpack在构建包的时候会按目录的进行文件的查找，resolve属性中的extensions数组中用于配置程序可以自行补全哪些文件后缀：

//webpack在构建包的时候会按目录的进行文件的查找，resolve属性中的extensions数组  
// 中用于配置程序可以自行补全哪些文件后缀：  
resolve:{  
 extensions:['\*','.js','.json']  
}

## externals

用于避免一些库的源码被构建到运行时文件中，但是又能用

//使用该库时  
externals: {  
 "jquery": "jQuery"  
},

## plugins

存放一些必要的插件

plugins: [  
 new webpack.HotModuleReplacementPlugin()  
],