# Webpack1.0

1、编译

①webpack main.js bundle.js

将文件main.js 编译为bundle.js文件输出；

②一般建立一个webpack.config.js文件来配置webpack



选择含义webpack.config.js文件的目录，运行命令：

②-1，webpack 开发环境下编译；

②-2，webpack -p 产品编译及压缩；

②-3，webpack --watch 开发环境下持续的监听文件变动开进行编译；

②-4，webpack -d 引入source maps

②-5，webpack --colors 显示静态资源的颜色；

②-6，webpack --profile输出性能数据，可以看到每一步的耗时；

1. 编译es6+jsx

首先，安装所需的模块（babel-loader babel-core babel-preset-es2015 babel-reset-react）：

配置方式一：



配置方式二：



注意：这里的exclude必须要有，表示node\_modules中的所有模块不需要用babel编译；

如果导入文件的时候想省略文件的扩展名，需要添加resolve.extensions来配置：



1. css样式和图片的加载

①加载静态资源



②配置

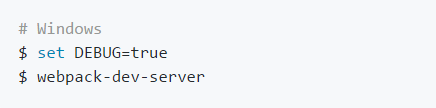


这里，如果图片小于8k这加载Data URL(base64),如果大于8k则不转化：



1. 环境切换（开发、生产）

命令行：

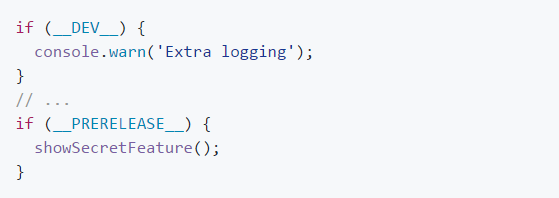


配置：



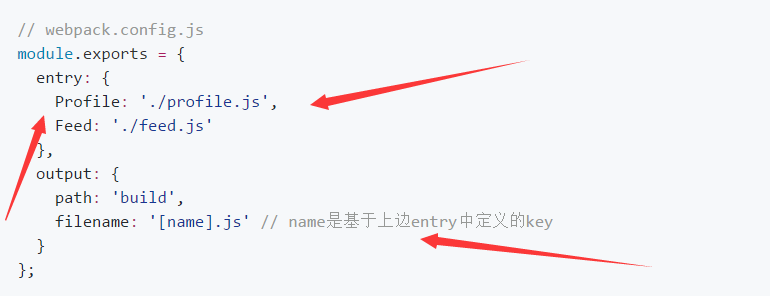
配置完成后，可以使用BUILD\_DEV=1 BUILD\_PRERELEASE=1 webpack 来打包代码；

代码中：



1. 多个入口文件

为每个页面创建自己的入口文件：



1. 提取公共代码（例如多个页面的公共react、样式和组件等等）





生成一个公共文件common.js

1. 异步加载

方式一：require.ensure，仅下载下来，引用该文件时，才执行了该文件内的代码



方式二：bundle-loader，需要下载该loader；



Webpack默认从项目的根目录引入这些异步加载的文件，可以通过配置publicPath来改变文件的引入路径：



1. 压缩插件（UglifyJs Plugin）

这是webpack自带的插件

配置webpack.config.js:



源代码：



经压缩处理后：



1. 把库文件提取到一个单独的文件保存

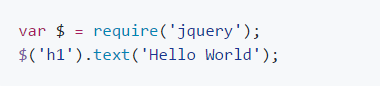
配置：



引用：



代码：



把jQuery变成全局变量，无需通过require来引用：



1. 热加载（webpack-dev-server）

方式一：命令行



--hot,仅更新组件被改变的部分；补充：如果配置了hot，则配置文件（webpack.config.js）不需要再配置HotModuleReplacementPlugin插件了；

--inline,刷新整个页面

--hot --inline,仅更新被改动的部分（HMR），如果失败则刷新整个页面

--host,设置外部访问的ip地址（例如：--host 10.137.2.28）

--history-api-fallback,去支持使用HTML5 history API（在react router中用到）

方式二：webpack.config.js配置



11、CSS Module

用法：

css文件：

.h1 {

color:red;

}

:global(.h2) {

color: blue;

}

js文件：

<div>

<h1 className={style.h1}>Hello World</h1>

<h2 className="h2">Hello Webpack</h2>

</div>

配置：

test: /\.css$/,

loader: 'style-loader!css-loader?modules'

12、加载第三方插件

配置：

var HtmlwebpackPlugin = require('html-webpack-plugin');

var OpenBrowserPlugin = require('open-browser-webpack-plugin');

module.exports = {

entry: './main.js',

output: {

filename: 'bundle.js'

},

plugins: [

new HtmlwebpackPlugin({

title: 'Webpack-demos',

filename: 'index.html'

}),

new OpenBrowserPlugin({

url: 'http://localhost:8080'

})

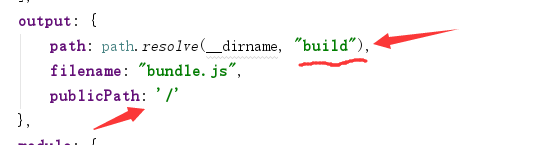
]

};

备注：具体有哪些插件以及用法，看官网；

1. publicPath

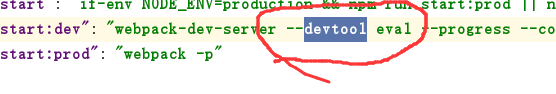
静态资源加载路径，用于生产环境；



这里的‘/’就相当于‘build’目录下；

1. 开发工具

命令行：



eval,开发环境，快速构建代码；