Webpack构建react开发环境

## 安装webpack

按照webpack主要通过node.js安装，node.js的安装可以参照教程

<http://www.runoob.com/nodejs/nodejs-install-setup.html>

进行安装（在命令行输入’node –v’已验证是否成功安装）

### 安装webpack：

npm install webpack –g

此处-g表示我们将在全局global安装webpack，这样就能够在任何路径下使用webpack了；

### 安装webpack-dev-server:

Npm install webpack-dev-server –g

Webpack-dev-server是webpack的web服务器，方便我们进行调试

### Webpack配置文件

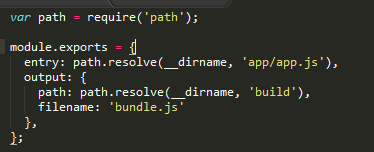
Webpack使用一个名为webpack.config.js的配置文件，在我们的项目的根目录下创建该文件；

设置我们的入口文件为app.js,位于app/目录下，并且希望webpack将它打包输出为build

/目录下的bundle.js，我们的项目文件结构为：

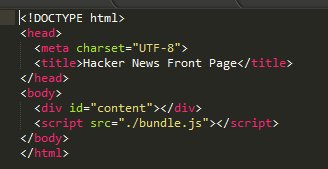


配置webpack.config.js



### 主页

在build/下创建index.html,内容为：



### App.js

在app/app.js编写测试代码：document.write(‘hello world!‘);

### Webpack打包

命令行进入项目根目录，执行webpack，即可进行打包工作：

### Webpack-dev-server运行

命令行进入项目根目录，执行webpack-dev-server，即可运行工程：

<http://localhost:8080/build/index.html>

## 配置package.json

在项目的根目录下命令行执行：

npm init –y  
会默认生成package.json

对其做部分修改，修改其script键值（快捷执行方式）为：



## 安装依赖

需要安装的依赖包括

1. React
2. Jquery（可选）
3. Babel
4. url-loader 、file-loader(用于模块化引入图片文件资源)
5. css-loader、style-loader（用户模块化引入css样式）

### 安装React：



其中—save表示所安装的依赖将会保存在本项目的根目录下的\node\_modules下

### 安装jQuery：



### 安装Babel相关依赖：



其中-dev表示开发环境

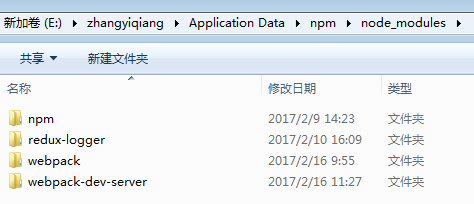
### 安装url-loader、file-loader:



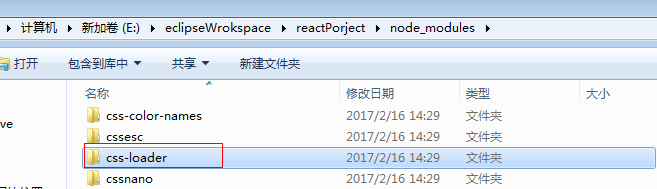
### 安装css-loader、style-loader:



安装上述依赖时应该确保网络畅通，除了webpack和webpack-dev-server是以全局安装，其文件放置在npm的应用文件下以外：



其余安装的依赖应该在项目根目录的\node\_modules有相应子文件夹，如：



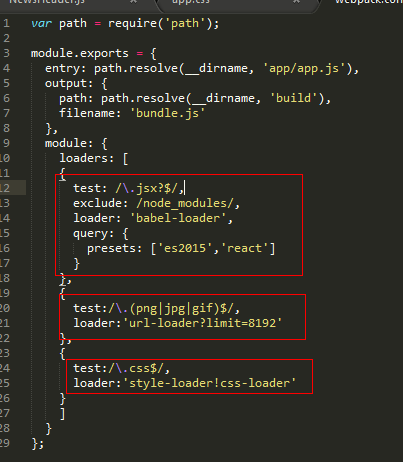






等等。

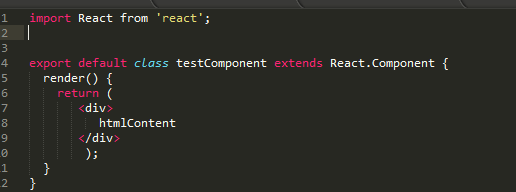
上述依赖安装完成之后需要在webpack.config.js中做相应配置，以便加载使用



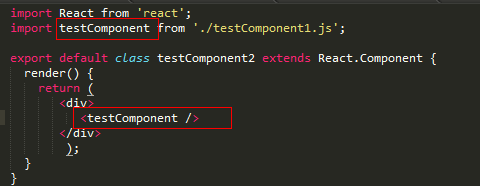
## 组件依赖引入

根据React组件化开发的设计模式，各个组件直接应该是隔离的互不相关的。每个组件应该以单独的js文件保存，通过导出（exports）的方式，由需要的组件通过依赖引入（import）

组件的js文件内容如下：



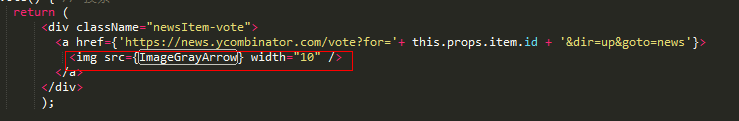
在需要引入的组件中引入就能够使用：



## 图片等文件资源模块化引入



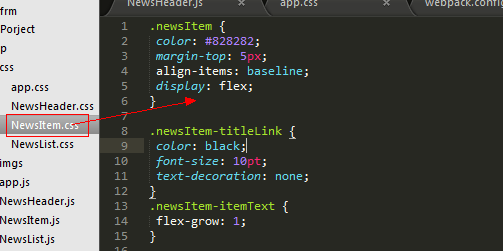
在需要使用到文件资源的组件文件头中引入，在该文件中就能够使用该资源：



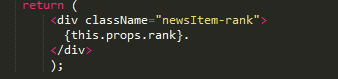
## 样式文件的模块化引入：



其中NewsItem.css的写法与常规css写法保持一致

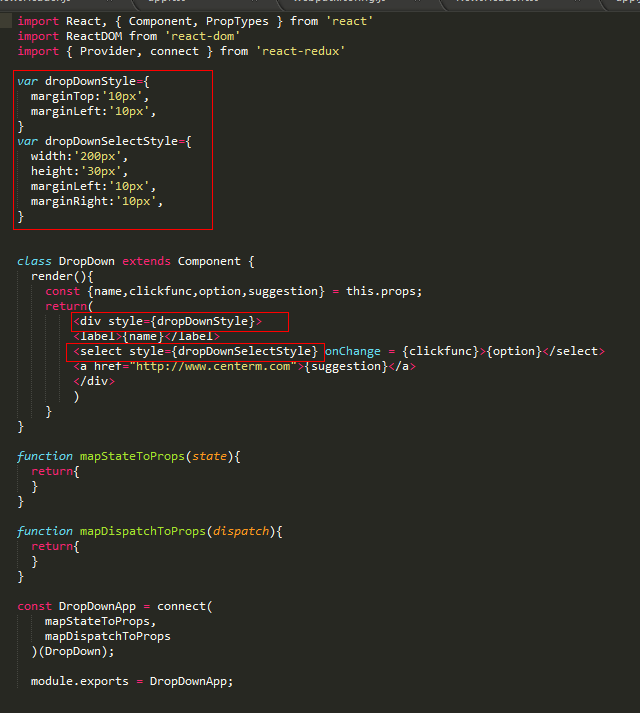


在组件中以className的方式直接引入样式



## 样式的另一种设置方式

在上述样式的引入方式中，与往常加载css文件一样存在着全局样式变量污染的问题，需要特别注意class名称的唯一性，否则会发生样式冲突、覆盖等问题。可以通过style属性来自动保持样式的唯一性：



如图：每个组件的样式以对象的方式保存，通过使用style={classObject}的方式可以保持保持样式的唯一性。

## 快速安装所有相关依赖

在项目根目录中，根据已有的package.json，命令行执行npm install，node将会自动下载所有相关依赖。package.json的内容如下（在手动安装时系统会自动生成）：

