# Webpack

## Webpack和grunt有什么区别

Grunt是一种能够优化前端开发流程的工具，而webpack是一种模块化的解决方案

Grunt的工作方式：在一个配置文件中，指明对某些文件进行类似编译，组合，压缩等人物的具体步骤，工具之后可以自动替你完成这些任务

Webpack的工作方式：把你的项目当做一个整体，通过一个给定的主文件（如index.js），webpack将从这个文件开始找到你的项目的所有依赖文件，使用loader处理它们，最后打包一个（或多个）浏览器可识别的javascritp文件

命令

webpack app/main.js -o public/bundle.js

以main.js为入口文件 找到main.js依赖的js文件，全部打包生成bundle.js文件

## Source maps

打包后的文件不容易找到出错的地方，对应我们写代码的位置，source maps可以帮助我们解决这个问题

Webpack可以在打包时为我们生成source maps 这为我们提供了一种对应编译文件和源文件的方法，使得编译后的代码可读性更高，也更容易调试

在webpack的配置文件中配置source maps 需要配置devtool

使用webpack构建本地服务器

Webpack提供一个可选的本地开发服务器，这个本地服务器基于node.js构建，不过它是一个单独的组件，在webpack中进行配置之前需要单独安装它作为项目依赖

npm install --save-dev webpack-dev-server

![]@WAQMOV~VMI4~M~MGR]((N](data:image/png;base64,)

在scripts中添加server属性 值为webpack-dev-server --open 用于开启本地服务器

## Loaders

通过使用不同的loader，webpack有能力调用外部的脚本或工具，实现对不同格式文件的处理，例如scss转为css 或者把下一代的JS文件转换为现代浏览器兼容的Js文件，对React的开发而言，合适的Loaders可以把React中用到的JSX文件转换为JS文件

Loaders需要单独安装并且需要在webpack.config.js中的modules关键字下进行配置

test:一个用以匹配loaders所处理的扩展名的正则表达式 （必须）

loader：loader的名称 （必须）

Include/exclude：手动添加必须处理的文件/文件夹或屏蔽不需要处理的文件/文件夹 （可选）

query:为loaders提供额为的设置选项（可选）

## Babel

Babel其实是几个模块化的包，其核心功能位于称为babel-core的npm包中，webpack可以把其不同的包整合在一起使用，对于每一个需要的功能或扩展，都需要安装单独的包（用的最多的是解析es6的babel-env-preset包和解析JSX的babel-preset-react包）

注意可能出现babel兼容性问题

Babel可以在webpack.config.js中进行配置，但是考虑到babel具有非常多的配置选项，在单一的weback.config.js文件中进行配置往往时文件显得太复杂，因此一下开发者支持把babel的配置选项放在单独的一个名为’.babelrc’的配置文件中进行配置(webpack会自动调用.babelrc里的babel配置选项)

## CSS

Webpack提供两个工具处理样式表，css-loader 和style-loader 二者处理的任务不同，css-loader使你能够使用类似@import和url(...)的方法实现require()的功能，style-loader将所有的计算后的样式加入页面中，二者组合在一起使你能够把样式表嵌入webpack打包后的JS文件中

在通常情况下css会和js打包到同一个文件中，并不会打包为单独的css文件，不过通过合适的配置webpack也可以把css打包为单独的文件的

## css module

Css module技术把js模块化思想带入css中来，通过css模块，所有类名，动画名默认都只作用于当前模块 只需要在css loader中进行简单的配置，就可以直接吧css的类名传递到组件的代码中，这样做有效避免了全局污染

## Css预处理器