**Webpack**

**Webpack是vuejs的一部分，首先学习webpack**

1. 首先全局安装webpack

npm i webpack -g

2、执行以下命令

Webpack --help

3、执行npm init

在目录下边产生一个package.json文件 里边有scripts文件 有项目依赖以及开发依赖

4、在本地安装webpack

npm i webpack -D（简写）

npm i webpack --save-dev（将模块加入到项目开发依赖）只—save（-S）放在项目依赖里

5、在项目中创建一个entry.js文件

6、模块打包（webpack 目标文件（入口文件） ）

webpack entry.js bundle.js

在文件中生成新的js文件，可以使用这个文件，创建一个index.html

Index.html引入生成的boundle.js文件，用浏览器打开

这个boundle.js是默认按照common.js规范做的封装，在浏览器资源看不到源文件

7、执行webpack --devtool 可以看到源文件

**Webpack启动服务server**

1、全局安装第三方组件npm i webpack-dev-server -g

2、在本地安装 npm i webpack-dev-server -D

3、webpack-dev-server –inline --hot（--inline实时监测修--hot对js修改）

在根目录创建一个webpack.config.js（配置文件）

1. 配置文件必须叫做webpack.config.js

**前端开发七大功能：**

一、模块化开发：

1、webpack天生支持commonjs

在vue下边创建src和prd（build/bundle）文件夹 创建commonjs和app.js实现模块化开发

1. webpack使用第三方组件，load加载器加载一个个模块

webpack（模块打包工具，实现前端工程化的所有功能）打包css（将css、img、html打包到bundle中） 需要装两个loader

npm i css-loader style-loader -D

1. webpack编译ES6

安装几个工具（babel的核心库、babel-lader、ES6的预设，才能run es6，用于webpack）

npm i babel-core babel-loader babel-preset-es2015 -D

这样才能在vue和react使用。

**babel的使用**

1. babel script.js -o script-bundle.js（文件级别的编译）

babel 要编译的文件名 -o（outfile输出文件） 输出文件名（在当前文件夹下产生）

1. babel src -d -o prd（文件夹级别的编译）

babel 要编译的文件夹名 -o（outfile输出文件夹） 输出文件夹名（在当该文件夹下产生）

以上命令可以在package.json文件中 scripts下边添加 “build”：“babel src -w -d prd ”

**npm run build**

在根目录下创建一个文件（.babelrc）

{

“presets:[“es2015”]”

}

执行npm run build

进行编译es2015（es6）

Babel装了预设es2015实现编译

安装react预设：cnpm i babel-preset-react -D

在.babelrc文件中

{

“presets:[“es2015”,”react”]

}

二、编译sass

1、安装sass-loader

npm i sass-loader -D

1. 安装插件

安装 node-sass

npm I node-sass -D

link引入css文件

安装extract-text-webpack-plugin

npm i extract-text-webpack-plugin -D

三、打包压缩

Plugins、进行新加

1、new webpack.optimize.UglifyJsPlugin()

补充：模块热替换（只要改代码，模块会自动编译，浏览器会自动刷新，使效率更高）

直接执行命令：webpack-dev-server –-inline –hot

另外一种方法：

在package.json中进行配置

“scripts”:{

“buile”:”webpack-dev-server –-inline --hot ”

}

执行命令：npm run build

入口命令：build 拿到项目，首先看package.jsonzhong 入口命令

2、修改端口号主机号等：（webpack.config.js）

3、工具类（mock是webpack的第五项功能）：

Json-server

全局安装json-server

创建mock文件夹 下有两个文件（mock.js/list.json）

然后启动json-sever执行命令json-server mock/mock.js

修改端口号

json-server mock/mock.js -p 9000

现在是另外启动一个服务，与webapck平级的服务，可以理解为阿帕奇服务，启动在端口号为9000，在全局装了一个json-server，靠json-server，可以启动无数个服务，只要端口号不同，这个服务给我们提供数据，相当于php。

ajax请求数据，

4、组件管理（webapck的第六项功能）

补充：版本号控制

总结：webpack的几大功能：

①写server，通过webpack-dev-server,

②模块化开发，自动生成文件

③编译scss

④合并打包

⑤mock数据

⑥版本号控制

⑦组建管理（npm）