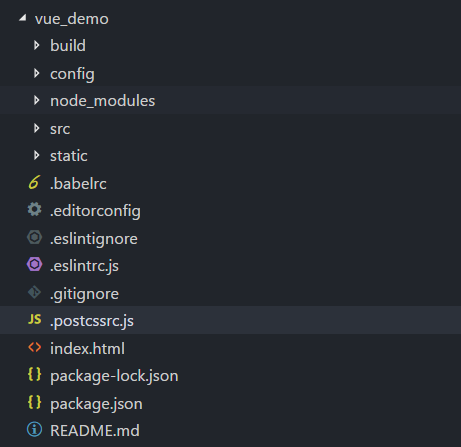
**Vue项目构建到打包体系**

整体结构：



配置文件夹build/config

config中

node\_modules依赖

src源码目录

static静态目录，放全局资源

babelrc：ES6转ES5/JSX转JS，rc：runtime control（babel的配置文件）

src目录下的main.js文件是程序入口文件，配置选项在webpack.base.conf.js（同样在其中配置了index.html，不可随意改动）

最终整个项目在index.html中打包生成了一个

<script type=”text/javascript” src=”/app.js”></script>

Main.js中的template: ‘<App/>’ （插件）会插入到el所匹配的页面上，此时渲染的根组件（根标签）就是App。（PS:组件就是一个局部的功能模块，可复用）

Src中目录中构建：

1. 构建入口main.js
2. 构建根组件App.vue

App.vue中的结构：

<template>

<div></div>

</template>

<script>

export default {

向外默认暴露一个配置对象

}

</script>

<style>

</style>

3．构建子组件（eg.HelloWorld组件）

<template>

<div></div>

</template>

<script>

export default {

配置对象必须写成函数

}

</script>

<style>

</style>

构建好子组件（HelloWorld），根组件来引用使用子组件（App.vue，三步：1.引入组件，2.映射组件标签，3.使用组件标签）,最后入口文件main.js引入使用根组件App.vue，创建Vue实例，（import Vue from ‘vue’ 注意大小写），el中的元素与index.html中的标签id对应，最重要将App.vue中的内容渲染到主界面中，先引入App.vue，再在components中将其映射成标签，最后使用组件标签。

template: ‘<App/>’

**Vue项目的打包发布：**