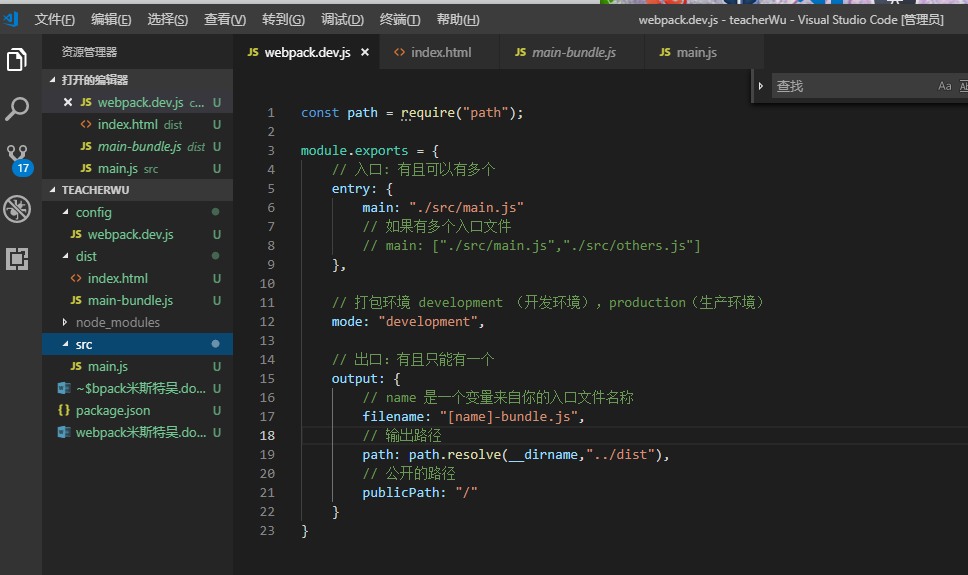
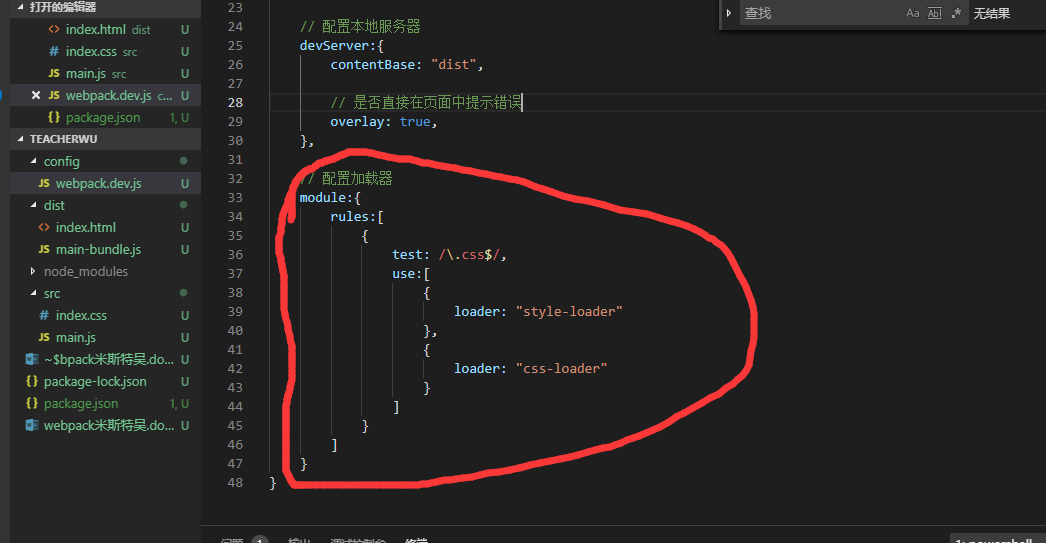
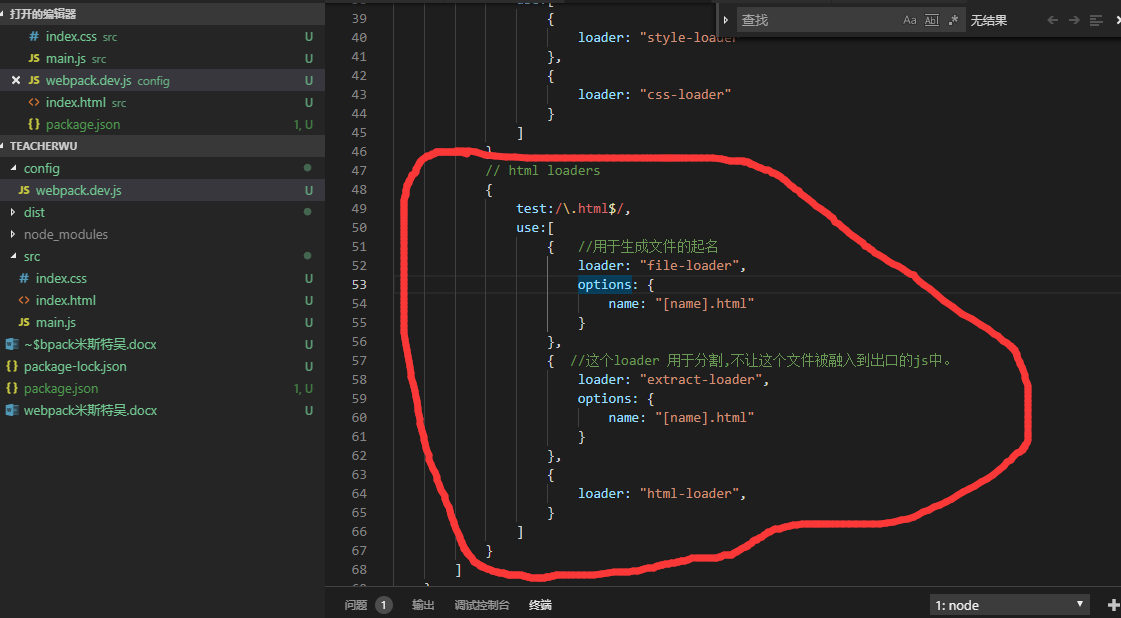
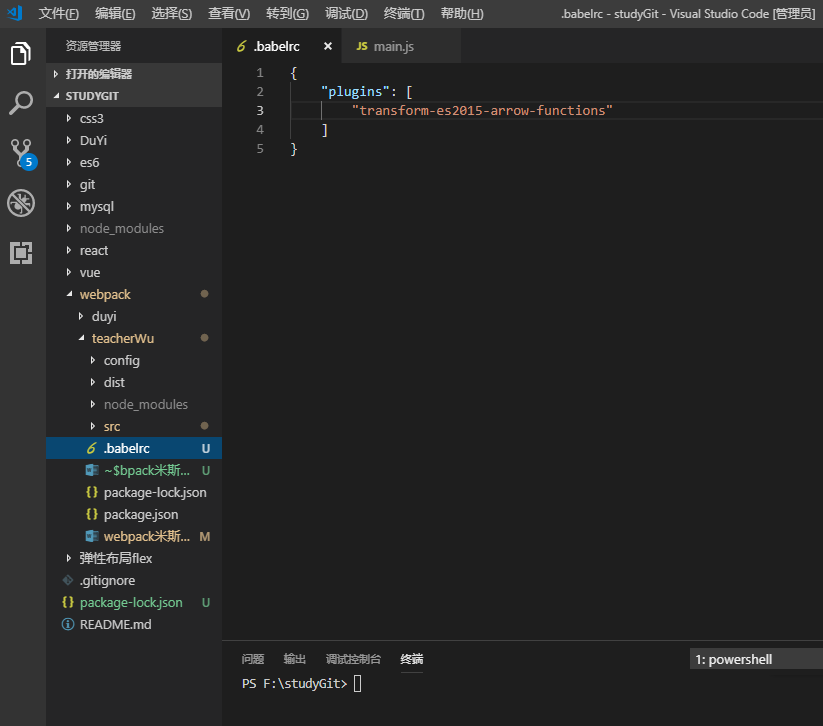
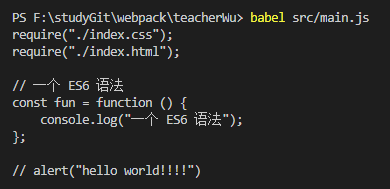
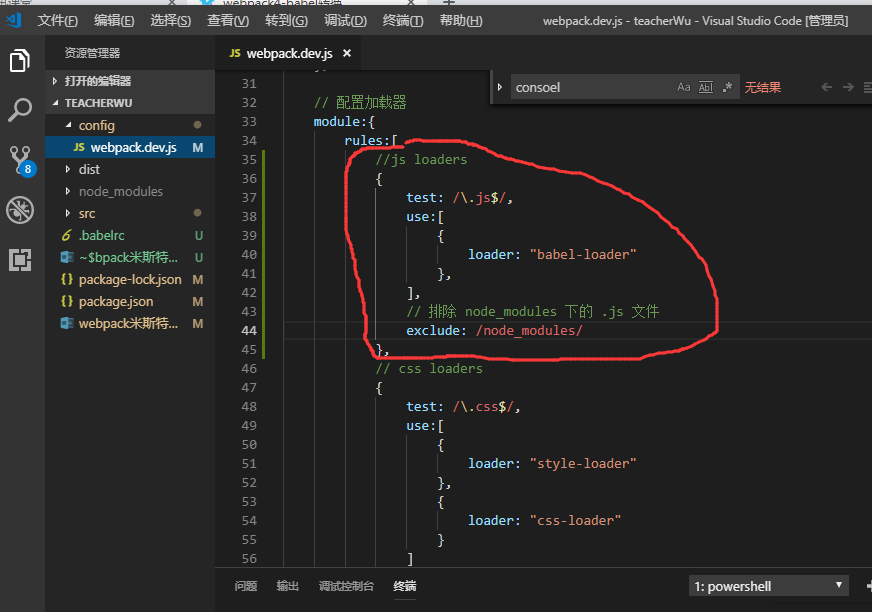
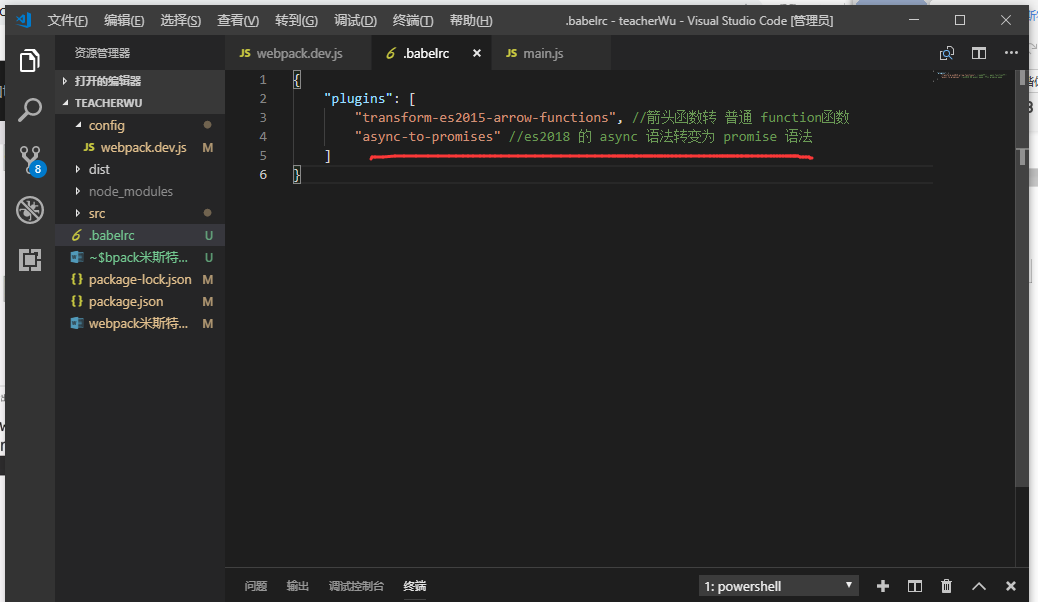
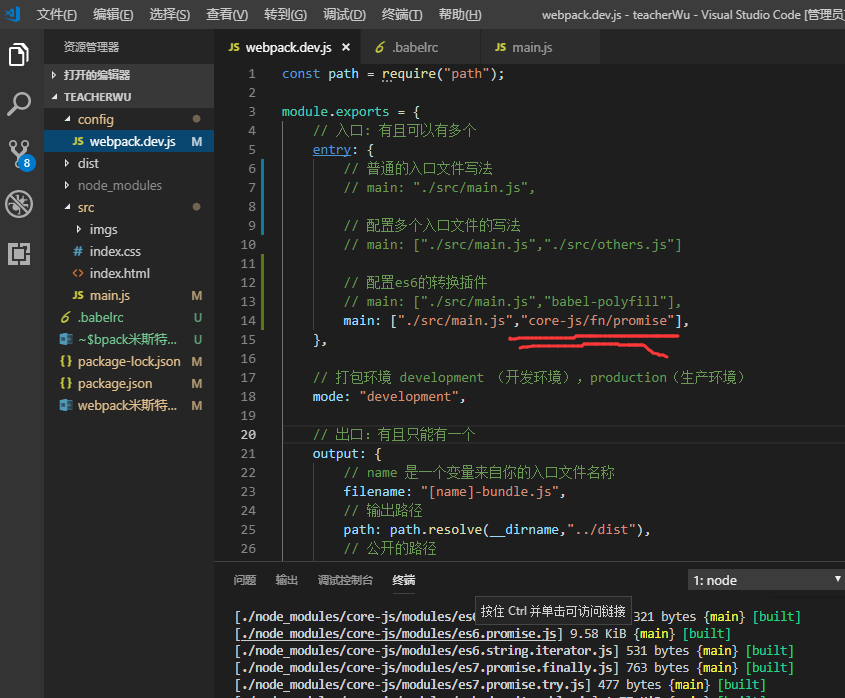
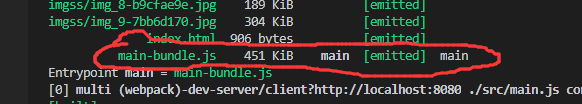
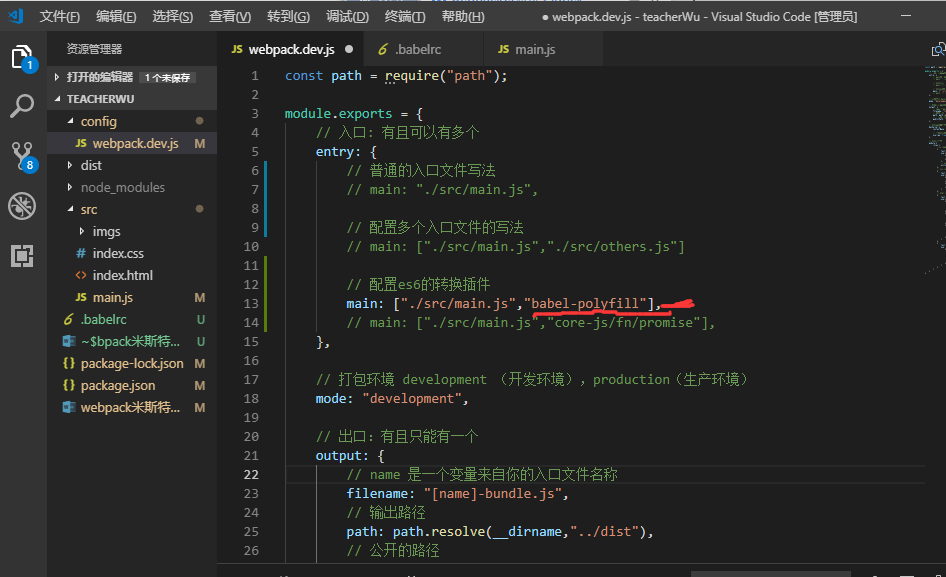
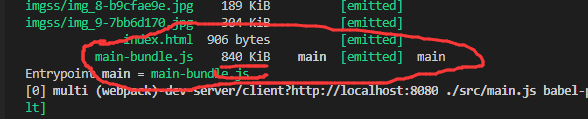
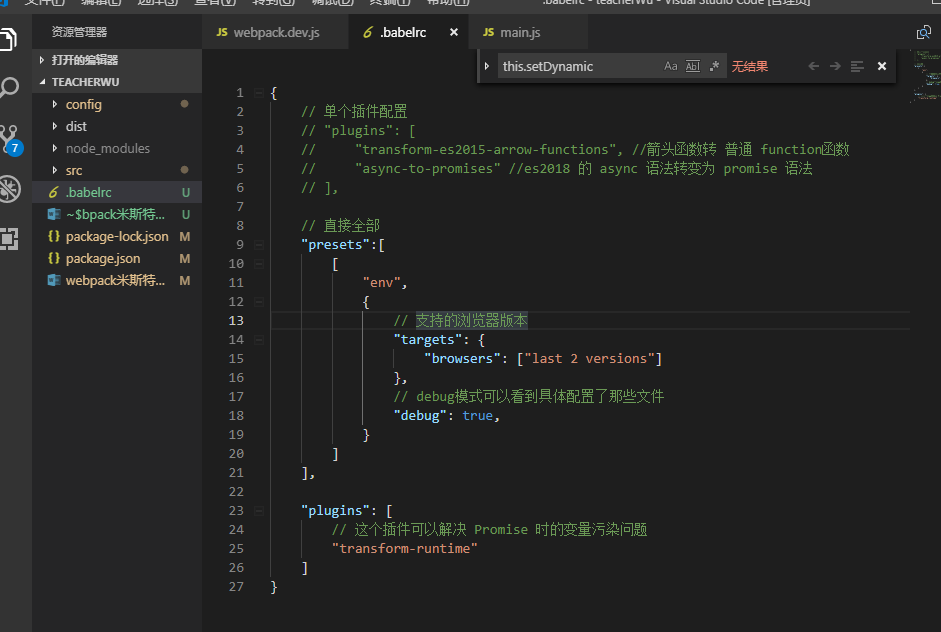
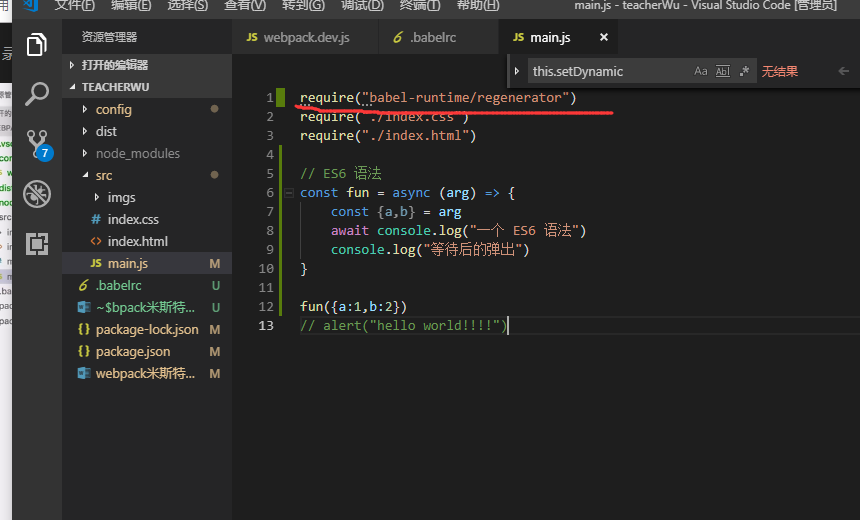
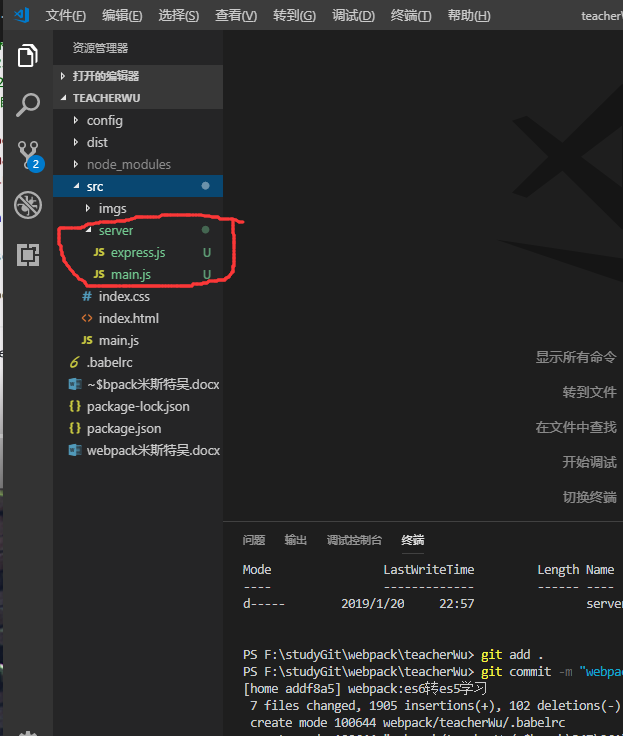
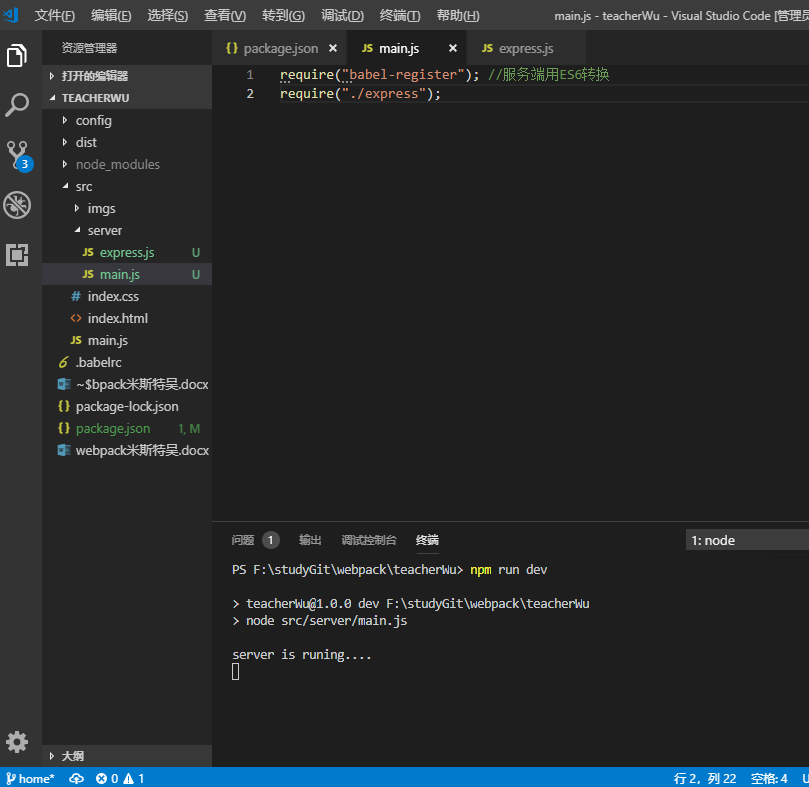
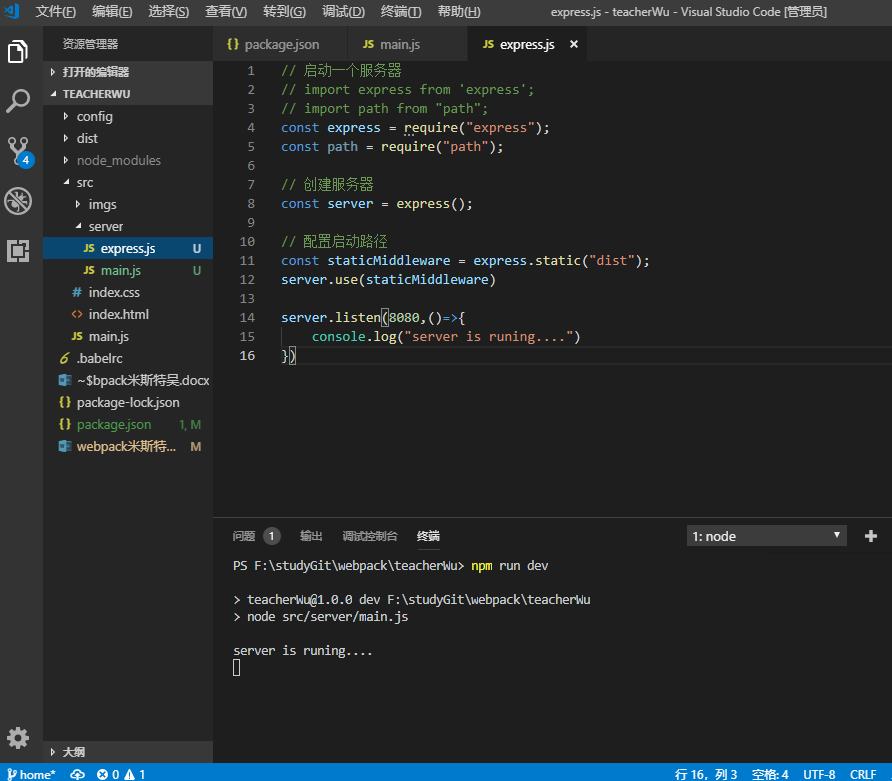
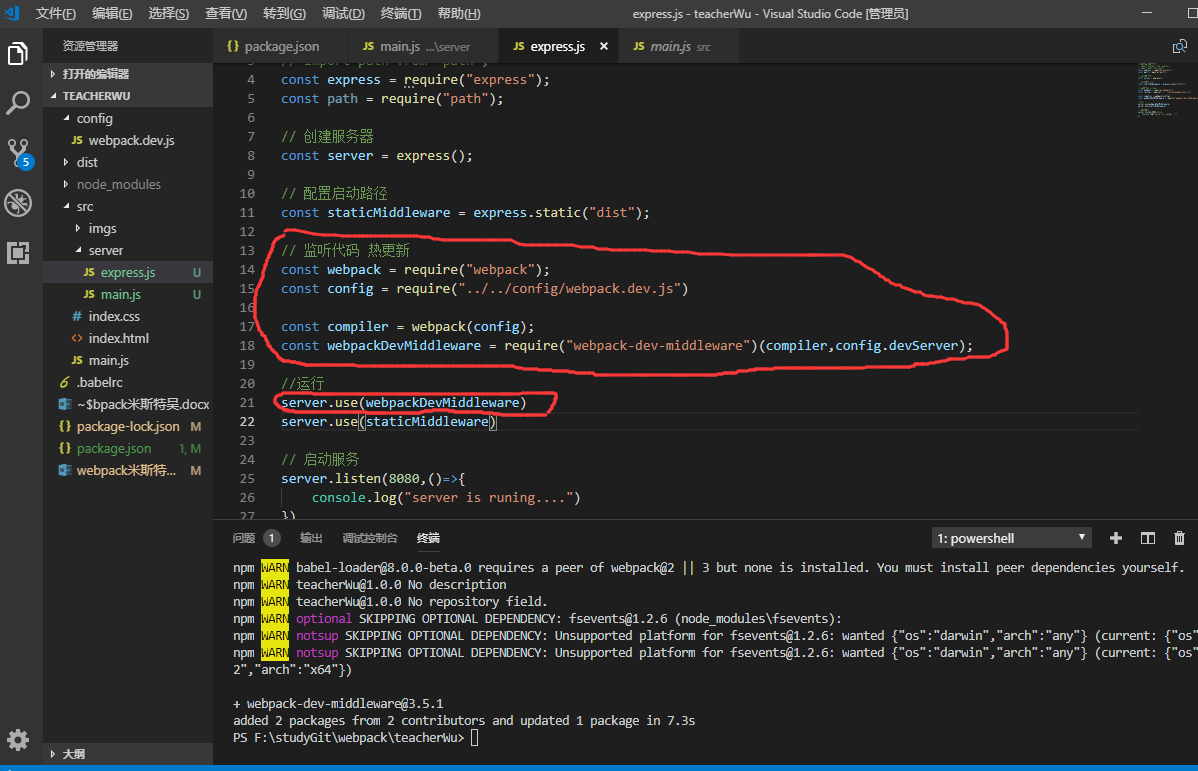
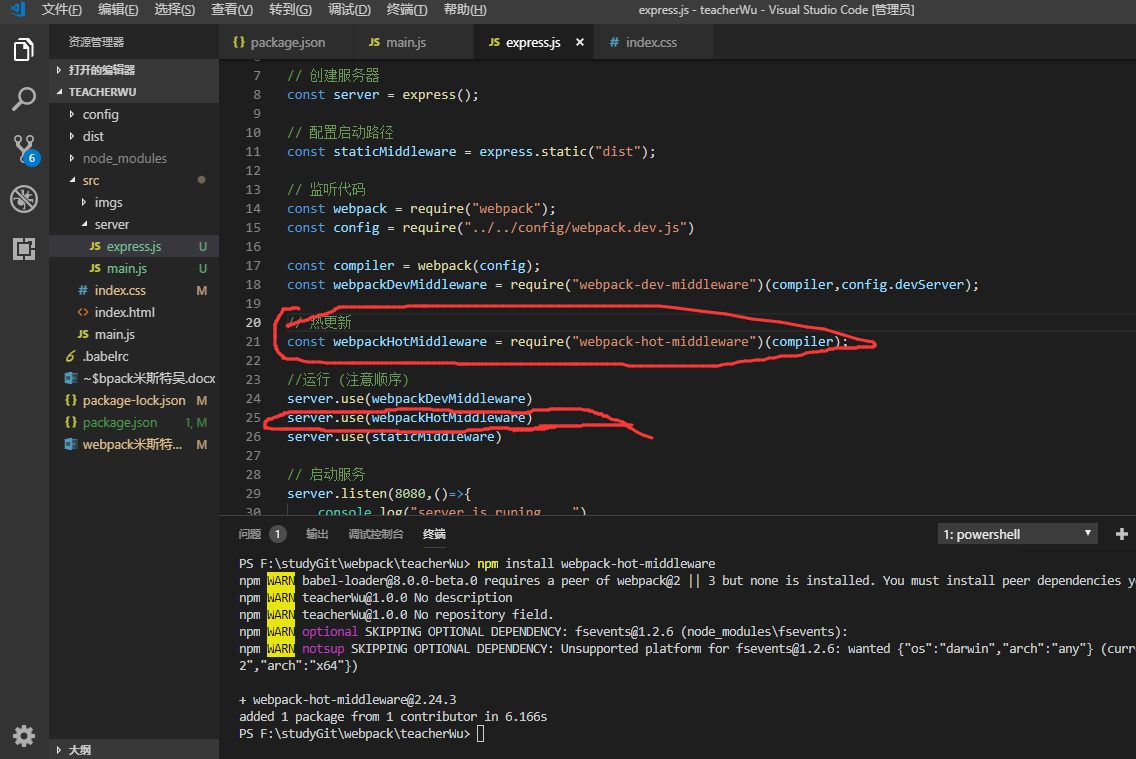
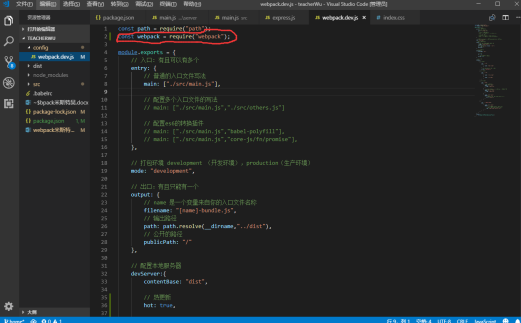
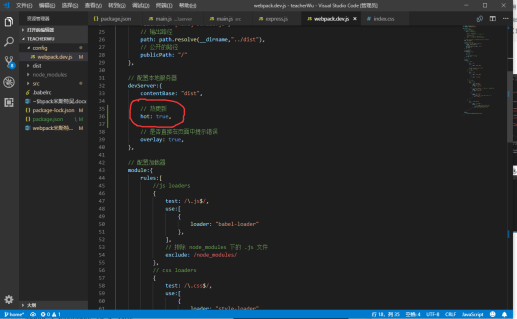
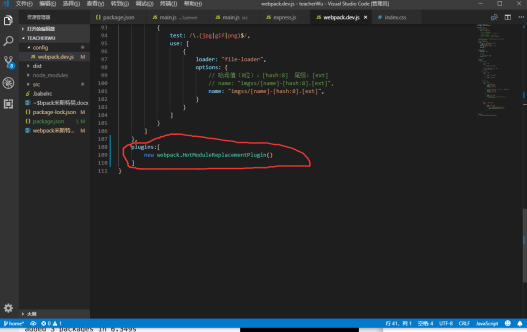
1. 安装 webpack webpack-cli webpack-dev-server （全局安装）  
   命令：npm install webpack webpack-cli webpack-dev-server -g
2. 创建3个文件夹 src ，dist ，config；然后初始化  
   命令：npm init -y
   1. Webpack4中有一个默认路径 src下的index.js可以直接使用 打包命令；  
      命令：webpack --mode=development （开发环境打包）  
      命令：webpack --mode=production（生产环境打包）
   2. 在config文件夹下创建 webpack.dev.js 文件配置文件  
        
      命令：webpack --config=config/webpack.dev.js
   3. 只要配置了 publicPath:”/”的路径，  
      可以直接在index.html中使用：<script src=”./main.bundle.js”></script>引入使用
3. 安装 webpack webpack-cli webpack-dev-server （本地安装）  
   命令：npm install webpack webpack-cli webpack-dev-server   
   1. 配置本地服务器在文件webpack.dev.js中  
      
   2. 本地安装后就可以启动**热更新**服务  
      命令：webpack-dev-server --config=config/webpack.dev.js
4. loader加载器（以为webpack只认识js所以加载其他文件必须使用加载器）
   1. 安装 .css 文件的加载器  
      命令：npm install style-loader css-loader  
      配置：  
      
   2. 安装 .html文件的加载器  
      命令：npm install html-loader extract-loader file-loader  
      配置：  
      
   3. 安装图片加载
5. ES6转ES5
   1. 单独使用babel 的不在webpack中
      1. 安装 babel  
         命令：npm install babel-core
      2. 安装 babel 安装一个能把**ES6的箭头函数转换为 ES5的函数**的插件  
         命令：npm install babel-plugin-transform-es2015-arrow-functions
      3. 创建es6的配置文件 .babelrs ，然后在里面配置插件  
         
      4. 全局安装 babel-cli  
         命令：npm install -g babel-cli  
           
         安装后可以直接使用   
         命令：babel [地址 例：src/main.js]  
         原文：  
           
           
         转换后：  
           
         
   2. 在webpack中配置使用babel
      1. 安装babel的loader  
         命令：npm install babel-loader  
         然后webpack中配置  
         注：这里直接派尔webpack 也需要安装   
          核心（npm install babel-core）和   
          需要的插件（babel-plugin-transform-es2015-arrow-functions）  
         
      2. 更好的转换最新的ES6语法 如：可以将ES2017的的async语法的转为 Promise实现  
         安装转换的插件  
         命令：npm install babel-plugin-async-to-promises   
         .babelrc中配置：  
         
      3. 安装最新的编译插件  
         命令：npm install babel-polyfill  
           
         在webpack.dev.js中的入口文件出配置
         1. 将babel-polyfill插件放在入口件  
            单个配置：  
              
            结果：  
              
            全部配置：  
              
            结果：  
            
         2. 使用环境变量在 .babelrc在中配置  
            安装：npm install babel-preset-env  
            配置：  
              
            其中配置的 transform-runtiome需要安装（这个插件可以解决放在环境变量的污染）  
            命令：npm install transform-runtime  
            然后在：main.js中加上一句  
            
6. 实现自己的脚手架
   1. 使用express创建服务  
      命令：npm install express  
      配置：在src下创建两个文件夹  
        
        
        
      但是不能热更新
   2. 安装webpack-dev-middleware可以帮助我们监听代码，能够自动编译到服务器  
      命令：npm install webpack-dev-middleware  
        
      这样每次修改文件，都会同步到服务器了，但是不能同步到浏览器，达不到热更新的效果，所以还需配置  
      安装：npm install webpack-hot-middleware  
        
      然后在config文件下webpack.dev.js 配置  
        
        
        
      src下最外面的那个mian.js中  
      