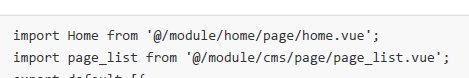
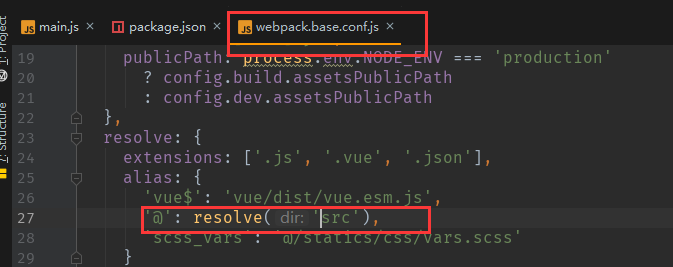
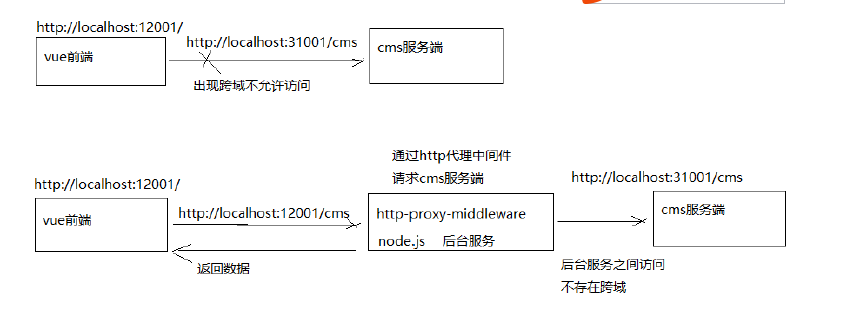
1：

@默认代表src

其实根据下面的配置得到的



2：vue中的使用代理解决跨域问题（基于node.js的）



使用代理中间件proxy，将请求转发给node后台服务，再有node后台服务请求cms服务，解决跨域，因为后台服务之间的请求不会产生跨域问题。

|  |
| --- |
| import http from '../../../base/api/public'  let sysConfig = require('@/../config/sysConfig');  let apiUrl = sysConfig.xcApiUrlPre;  export const page\_list = (page,size,params) => {  // return http.requestGet('http://localhost:31001/cms/page/list'+page+'/'+size);  //因为当前基于node服务进行前端开发，访问地址：http://localhost:11000/#/cms/page/list，如果这里直接请求cms后台服务，很明显会  //产生跨域问题  //解决方案，使用vue脚手架的代理中间件  return http.requestGet(apiUrl+'cms/page/list'+page+'/'+size);  //apiUrl就是一个常量值 = api，这样当请求的就是是/api/cms/page/...  //如果通过这个请求访问到我们cms后台服务呢？请看config/index.js的proxyTable配置，这个就是代理  /\*可以看到里面配置了  \* '/api/cms': {  target: 'http://localhost:31001',  pathRewrite: {  '^/api': ''  }  说明：当请求路劲是/api/cms打头的，我们就转发到target的url上面，并且将路劲和参数带去:  =》 http://localhost:31001/api/cms/list/...,但是我们后台请求路劲不包括api，所以pathRewrite  \* 就把/api用''替代了 = 》 http://localhost:31001/cms/list/...  \* \*/ |

