

问题一：如何解决跨域问题？

不要只说个jsonp，实际开发中 第二种方式用的多。

- 1、vue-cli项目中通过node.js代理服务器来实现跨域请求
- 2、在服务器响应客户端的时候，带上 Access-Control-Allow-Origin:* 头信息 [推荐使用]
- 3、通过 jquery 的 jsonp 形式解决跨域问题

详解：

- 1、在vue-cli项目中的 config 文件夹下的 index.js 配置文件中，配置 dev对象的 proxyTable对象，可通过node.js的代理服务器来实现跨域请求。

```
dev: {  
  env: require('./dev.env'),  
  port: 8080,  
  autoOpenBrowser: true,  
  assetsSubDirectory: 'static',  
  assetsPublicPath: '/',  
  proxyTable: {},  
  cssSourceMap: false  
}
```

但是这种解决方案，仅适用于 vue-cli 开发环境 解决跨域问题，生产环境如果涉及到跨域问题，并未解决

- 2、以 java 代码为例，设置 http 请求的响应头，推荐使用：

```
public class testFilter implements Filter {  
  public void doFilter(ServletRequest request, ServletResponse resp,  
    FilterChain chain)  
    throws IOException, ServletException {  
    HttpServletResponse response = (HttpServletResponse) resp;  
    response.setHeader("Access-Control-Allow-Origin", "*"); //解决跨域访问报错  
    response.setHeader("Access-Control-Allow-Methods", "POST, PUT, GET,  
      OPTIONS, DELETE");  
    response.setHeader("Access-Control-Max-Age", "3600"); //设置过期时间  
    response.setHeader("Access-Control-Allow-Headers", "Origin, X-Requested-  
      With, Content-Type, Accept, client_id, uuid, Authorization");  
    response.setHeader("Cache-Control", "no-cache, no-store, must-  
      revalidate"); // 支持HTTP 1.1.
```

```
response.setHeader("Pragma", "no-cache"); // 支持HTTP 1.0.  
response.setHeader("Expires", "0");  
chain.doFilter(req, res);  
}
```

3、jQuery设置ajax请求跨域:

```
$.ajax({  
    url:'...../data.js',//可以不是本地域名  
    type:'get',  
    dataType:'jsonp',    //jsonp格式访问  
    jsonpCallback:'aa'   //获取数据的函数  
})
```