1.网上的资料参差不齐，很多资料的说法是过时的，甚至有的说法根本就是自身理解错误的，比较容易让人产生误导。尽信书不如无书，绝知此事要躬行。时间是检验真理的唯一标准，要结合自己的实践辩证的看待网上和书上的资料，相信自己相信实践，同时要深入分析。

2. 解决跨域问题的两种方式：

(1).CORS.

(2).Nginx.

3.下面介绍CORS方式。

所谓跨域，针对的是不同域名下的前后端服务，前端通过Ajax不能跨域访问后台服务的问题，首先浏览器就知道你ajax的请求是不是一个跨域请求，因为当前的资源文件（html，js，css）是请求了某一个域名返回到浏览器的，浏览器自然知晓这些资源的域名是什么，当这些资源

的js发起ajax请求时，请求的域名与其所在的资源的域名是否一致浏览器当然是知道的。

跨域问题产生的根源是浏览器跨域访问策略问题，早起的浏览器根本就不允许跨域访问，只允许同域访问，要解决跨域访问非常复杂，比如JSONP方式，需要前后端都要做实现，并且最初还只支持GET请求，后来经过改进，可以支持POST方式了，也是非常复杂。

随着社会的发展和科技的发展，各浏览器厂商意识到，不支持跨域访问是多么what a fuck的事情，但是如果没有任何限制的支持又会轻易的造成安全问题。所以为了简化跨域问题的解决，浏览器厂商的后续版本中就修改了他们的跨域策略，就有了CORS这个W3C标准，

也就是浏览器官方支持的一种解决跨域问题的方式，这种方式相比JSONP方式，确实简单了不少，只需要后台服务器给予header响应即可，浏览器根据服务器的响应，即可知晓服务器的意图，即服务器对本次请求的域名是否“认可”，如果认可，那么浏览器就放行响应体，

如果“不认可”，那么就对响应数据进行劫持，并报出js错误，即浏览器就知道了，服务器不放行本次跨域请求。只是使用CORS方式解决跨域问题，其他一些问题依然寸，虽然不是它造成的。

CORS是一个W3C标准，全称是"跨域资源共享"（Cross-origin resource sharing）。它允许浏览器向跨源服务器，发出XMLHttpRequest请求，从而克服了AJAX只能同源使用的限制。

CORS需要浏览器和服务器同时支持。目前，所有浏览器都支持该功能，IE浏览器不能低于IE10。

整个CORS通信过程，都是浏览器自动完成，不需要用户参与。对于前端开发者来说，CORS通信与同源的AJAX通信没有差别，代码完全一样。

因此，实现CORS通信的关键是服务器。只要服务器实现了CORS接口，就可以跨源通信。

如果服务器不做任何事情，那么当前端发起AJAX请求时如果是跨域访问的，无疑会存在跨域问题。

问题的现象是请求可以正常发送到服务端，但服务端的响应数据会被浏览器拦截，浏览器拦截到响应的头部数据后，发现服务器返回的响应头信息中并没有返回支持跨域的header信息，因此认定服务器不支持本次的跨域请求，所以对于响应数据浏览器也做了劫持，不予响应，同时报出js错误。

对于CORS方式，常见的就是本地前后端联调时经常用到，非常方便，一般线上不需要使用这种方式解决跨域问题。原因在于，一个前端项目通常可能会有几个后端项目做支持，如果一个前端项目的页面中需要跨域访问多个后端项目进行数据请求，那么首先存在的一个基本问题就是cookie的问题，

尤其是几个后端项目可能存在主域名都不同的情况，这时，session会话就不太容易做了，因为不管是传统session会话还是自实现的redis的session会话，一般都需要cookie的配合才会比较容易或规范的实现。如果出现了跨域问题，那么cookie信息就不在同一个域里，会话保持就是一个问题，如果

用CORS这种方式解决了跨域问题，那么对应的session会话问题，可能就要考虑单点登录等进行解决了，也就是说CORS可以解决跨域问题，却不能解决会话问题。

可能会说单点登录也可以啊，是的，是可以，但是通常单点登录是针对完全不同的系统之间进行的，现在的这几个后端项目虽然是几个项目，但应该属于同一个大系统的后端微服务，完全没有必要使用单点登录。不是不能，而是不适合，徒增系统的复杂度。

4.下面介绍Nginx方式。

Nginx是HTTP和反向代理服务，通常用作前端ui的静态资源服务器和后端请求的反向代理服务器。使用Nginx可以完美的屏蔽跨域问题，并且是我们测试环境和线上真实使用的一种方式。

把前端ui项目部署在Nginx服务器上，那么该ui资源（html,js,css）的获取都是该nginx服务器的域名作为主域名进行请求的，而前端ui项目中所有的AJAX请求的域名也都是以该Nginx服务器的域名作为主域名进行请求的，因此属于同域访问，自然不存在跨域问题。

至于AJAX请求，会通过Nginx主域名后的二级路径由nginx进行反向代理服务分发，最终请求到对应的后端项目tomcat服务器上，而从nginx到后端项目tomcat服务器的过程是属于后台http请求，与浏览器无关，不受浏览器的跨域限制，因此天然不存在跨域问题。