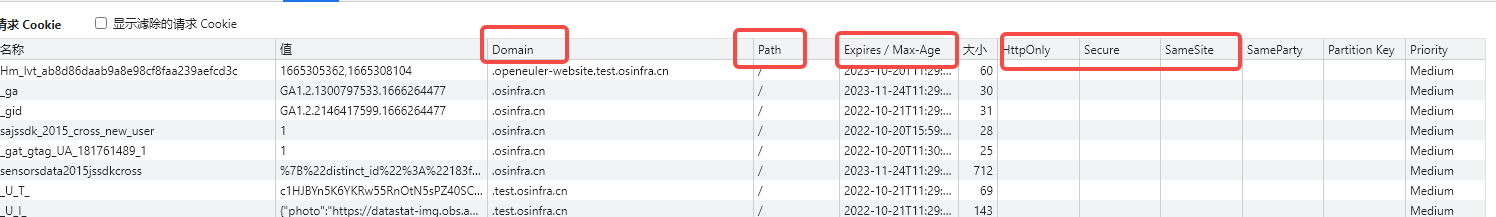
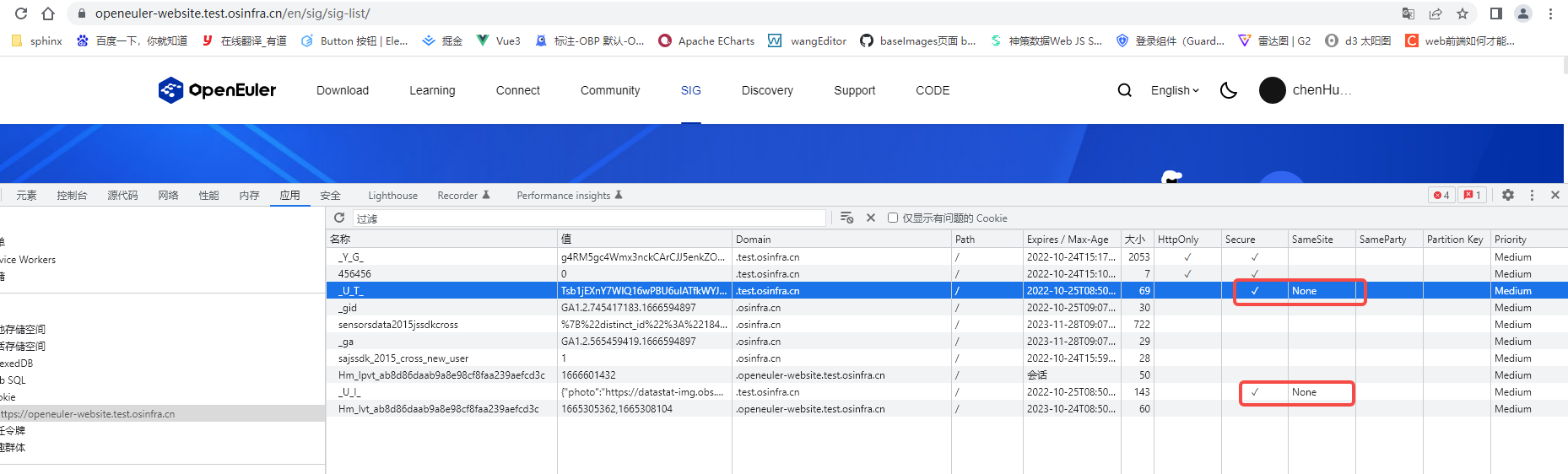
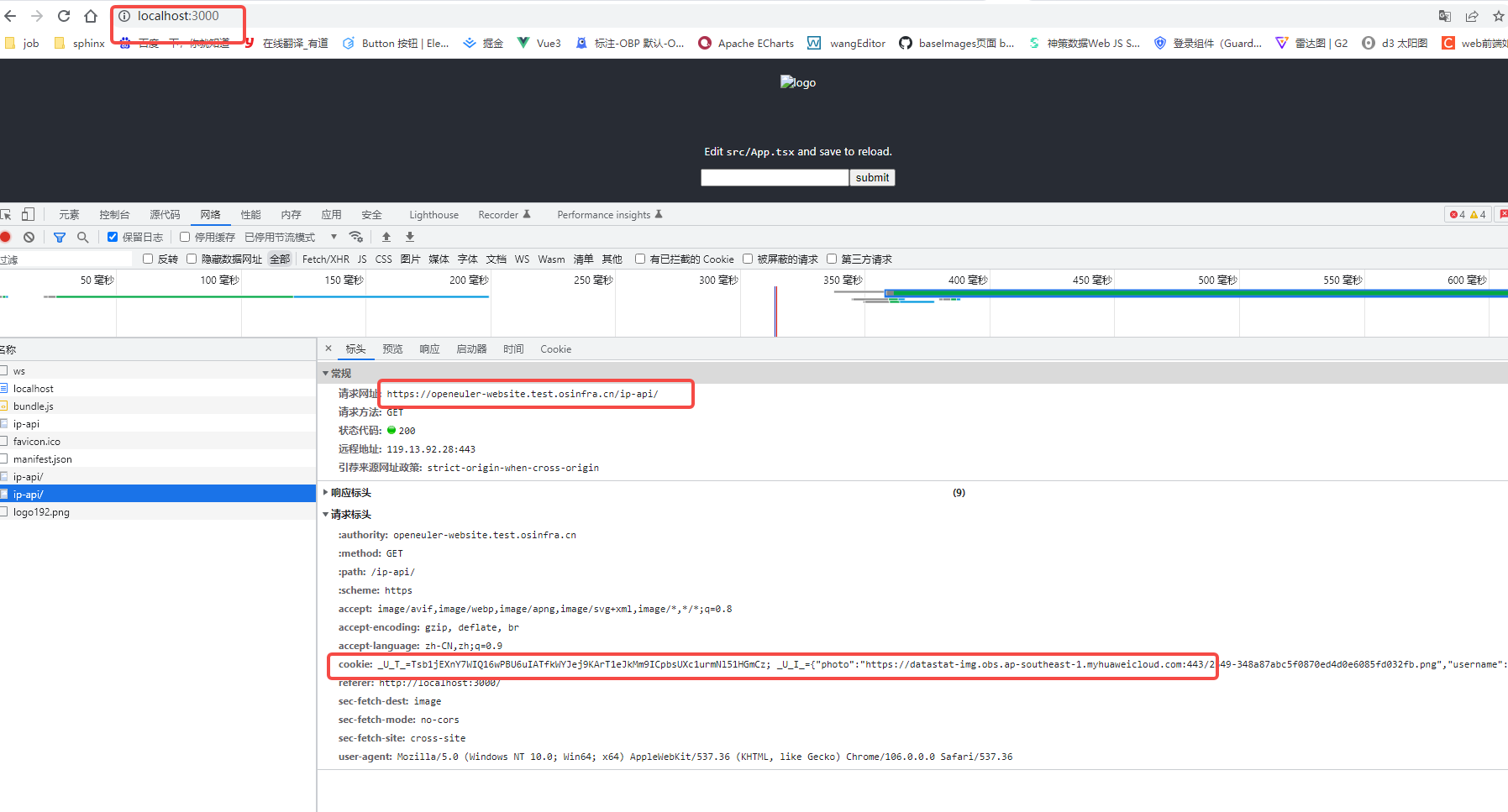
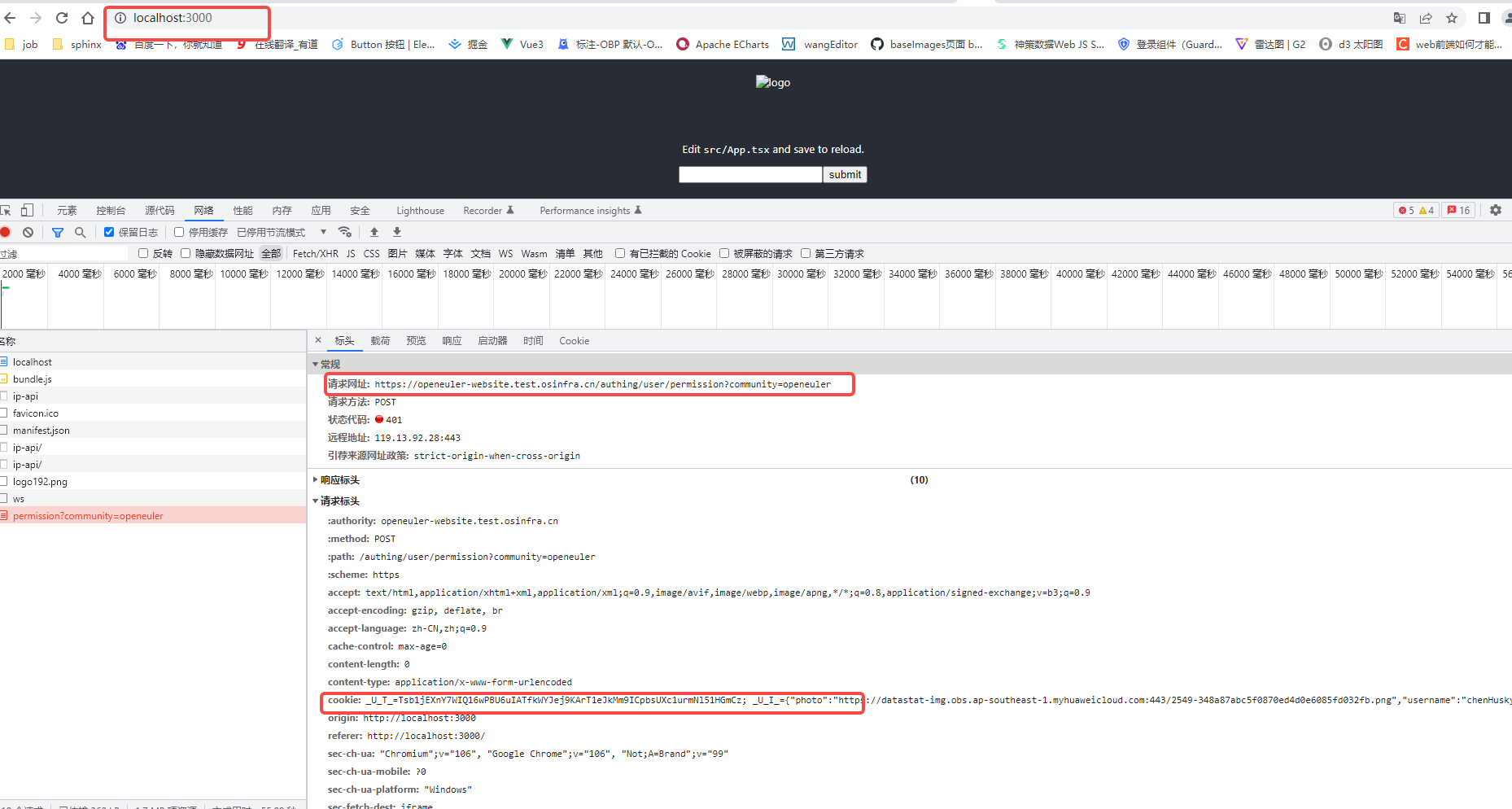
1. 背景  
   在openeuler官网使用单点登录时，在保存token通过设置cookie 的 domain属性来保证各个域名（子域相同 .test.osinfra.cn）下登录状态保持统一
2. Cookie介绍  
   1.cookie是客户端的解决方案，是服务器发给客户端的特殊信息，这些信息以文本文件的方式存放在客户端，后续请求，客户端在请求头都会带上这些特殊的信息。  
   2.cookie重要参数介绍：  
     
   ①Domain: 设置该条cookie信息基于域名的作用域。当前项目将token存储到cookie时，设置domain为.test.osinfra.cn，此时openeuler-website.test.osinfra.cn与openeuler-doc-zh2.test.osinfra.cn均能读取到该token，保证了相同浏览器下这两登录状态的一致  
   ②Path: 设置该条cookie信息基于路劲的作用域。一般默认设置均为 / ，表示全路径下均能访问该cookie；若设置为/zh, 则/en 路径下则不能访问，反之亦然  
   ③Expires/Max-age: Expires和Max-age均为Cookie的有效期，Expires是该Cookie被删除时的时间戳，格式为GMT,若设置为以前的时间，则该Cookie立刻被删除，并且该时间戳是服务器时间，不是本地时间！若不设置则默认页面关闭时删除该Cookie。Max-age也是Cookie的有效期，但它的单位为秒，即多少秒之后失效，若Max-age设置为0，则立刻失效，设置为负数，则在页面关闭时失效。Max-age默认为 -1。  
   ④httpOnly: 设置为true时，不允许js脚本通过document.cookie读取该值，但发送请求依然会发送该值  
   ⑤secure: Secure为cookie的安全属性，若设置为true，浏览器只会在https和ssl等安全协议中传输该值，http则不传送（ps：设置为true时，在http下浏览器中也无法查看该值）  
   ⑥sameSite: 属性可以让 Cookie 在跨站请求时不会被发送，从而可以阻止跨站请求伪造攻击（CSRF），他有三个值  
   ·Strict 最严格，完全禁止第三方，跨站点时，任何情况都不会发送cookie  
   ·Lax 规则稍宽，大多数情况也不发送，但是导航到目标网址的get请求除外  
   ·None 关闭该属性，但必须设置Secure为true  
   注: 同站是指有效顶级域名+二级域名相同，不考虑端口与协议；如www.a.com与www.b.com为跨站,openeuler-website.test.osinfra.cn与openeuler-doc-zh2.test.osinfra.cn为同站；其中osinfra.cn为顶级域名，test为二级域名
3. 当前项目使用方案  
    项目中需要将token等登录信息存放在cookie ,并设置domain做到各网站登录状态统一；  
   cookie中存储了两个token，每次请求时做双token验证，并同时后端对请求头的referer字段做白名单校验  
   1. 后端在cookie 设置token并设置httponly，该token含有用户信息，前端全程不感知，可  
   防止xss攻击通过document.cookie获取token  
   2. 前端获取另一个token（这里为了区分第一条的token，以下称headerToken），headerToken是登陆成功后后台返回的，headerToken不含用户信息，使用时放在请求头里；该对策是为了应对csrf攻击，是第一层防护的补充  
   3. 对请求头referer字段做白名单限制，可进一步加强第三方页面对我们接口请求的限制
4. 简单介绍下csrf  
    CSRF(Cross—Site Request Forgery) 跨站请求伪造。就是第三方网站在调用我们接口时，浏览器自动带上了cookie信息去发送请求，第三方是不感知我们的cookie的，只是利用而已。  
   简单的举个例子说明：当你登录成功我们网站后，点击了钓鱼链接，跳转到了第三方，第三方网址使用img 或者form表单方式发送用户信息接口，如果值只验证cookie里的token，接口就会请求成功，那么这个用户信息就可以被第三方知道，所以我们需要再验证header里的token  
     
   目前谷歌浏览器给出了个解决方案，就是设置samesite为lax或strict（谷歌默认是lax），但是这个属性是比较新的属性，有兼容性，火狐浏览器默认设置为None（但测试火狐也不会被img，form跨站携带cookie，估计有其他处理）  
     
   接下来简单实现下csrf携带cookie发送请求  
   1. 先把samesite设置为None（因为新版的谷歌浏览器已经自己处理过这个漏洞咯，但不保证用户用的旧版或者ie？？？？）  
     
   2. 写个form、img请求接口代码  
     
   3. 启动项目，看看效果  
   img的：  
     
   form的：  
     
     
   安全问题无小事！