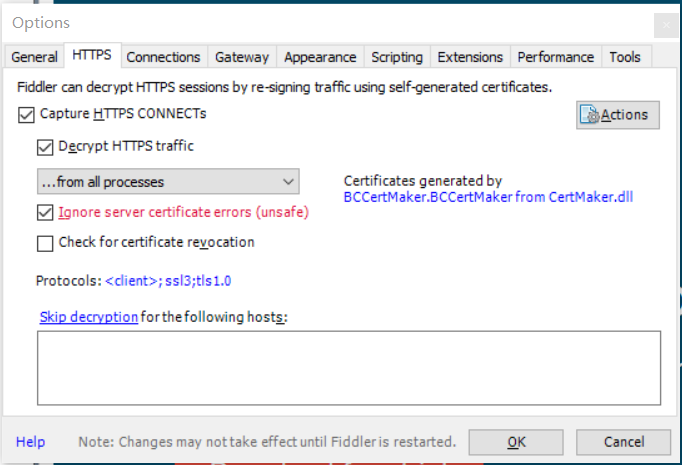
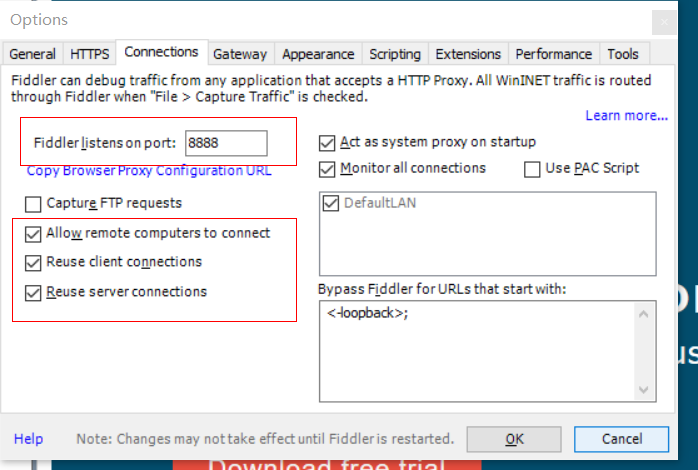
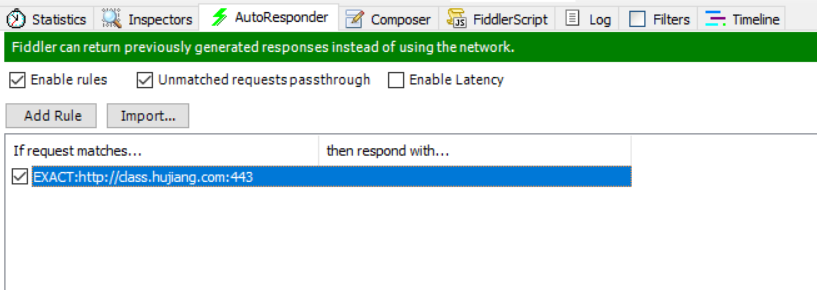
https请求允许



设置代理端口，允许第三方设备连接本地电脑作为代理



设置本地文件代理线上js文件



4.最后在自己手机设置【最底部的HTTP代理】，代理到自己电脑的ip【电脑用ipconfig找出局域网的ip地址：注意手机和电脑要在相同局域网；如果电脑是网线连接上网的，则需要在这个局域网下找一个wifi，连接此wifi】，

端口设置8888，这样自己手机访问对应页面请求，会通过自己电脑来代理，相当于自己电脑发送请求，自己电脑又被fiddler代理，fiddler就捕获了对应的请求，然后用自己电脑的js代理。（需要在自己手机输入ip：端口）然后下载fiddler的证书，这样才允许访问



注意：

1. 注意手机和电脑要在相同局域网，如果电脑是网线，那么手机需要在电脑的局域网下找一个wifi，然后连接这个wifi【或者手机开热点，电脑连接手机热点应该也可以尝试，虽然没试过】
2. fiddler的端口8888和自己手机设置的端口要相同，都是8888
3. 一般Hybrid包是会隐藏报文请求，需要app做相应开发测试app，允许拦截报文
4. 如果是qa上有问题需要用本地文件模拟，前面设置都一样，同时在本地启动node的html服务和静态资源服务，然后在fiddler中用localhost的静态资源来代替qa的静态资源（这样能利用本地开发的热替换，本地js改完以后，localhost会自动更新最新，然后再手机浏览器刷新页面，也能获得最新js，因为目前开发的js，都需要babel编译才能在浏览器执行）
5. 手机上，必须安装fiddler的证书（证书下载地址：代理电脑的IP：端口），要下载证书，安装证书，设置信任证书
   1. 手机浏览器输入“代理电脑的IP：端口”，出现上面第4步界面，点击下载证书，就是上面4中的下载证书链接
   2. 下载好以后，需要在“设置”的“现有IOS15.7.1”可用【右侧有红圈圈】，点击去，可以看到对应的fiddler的证书，在这里进行证书安装，但是安装好以后还不行，还需要设置信任该证书；
   3. 需要在“设置”>通用>关于本机>证书信任设置 把证书设置成完全信任

不设置完全信任的话，网站报文一样抓不到

1. 如何在手机上访问本地开发地址，实现手机上调试：

只要在wifi上设置本地开发地址，然后再手机上输入本机的开发地址即可

如果想抓本机的报文，就可以直接

然端口设置8888，这样fiddler就能抓取报文了，

1. 最最重要的一点：个人电脑连接的网络，可以在“防火墙和网络保护”【搜索防火墙和网络保护】中看到，需要取消勾选“阻止所有传入连接，包括位于允许应用列表中的应用”，这个是大坑
   1. 自己在使用哪一个网络：域网络，专用网络，公用网络【使用中】，可以看出自己在使用公用网络，点开对应自己使用的网络，如果下面一旦勾选了“阻止所有传入连接，包括位于允许应用列表中的应用”，那么手机上就无法使用“计算机代理”手机的请求了，因为手机的代理请求会直接被计算机的网络阻止。

1.Hosts+nignx通过反向代理来实现本地开发的跨域问题：【需要后端只是，很多前端都不会，不推荐】

Hosts文件可以把127.0.0.1设置为本地的开发地址，这样访问网站，就不需要使用localhost:path/…;而是直接用开发地址dev.baidu.com/path/….;可以解决本地开发的“跨域的问题”，hosts文件修改，可能会用到nginx的反向代理！！！！！！！！！！！！！！

Hosts文件，你可以把他看出一个dns服务器，用于解析域名到制定的ip，比如设置127.0.0.1 [www.baidu.com](http://www.baidu.com) 那么浏览器里面请求[www.baidu.com,就会自动被映射到127.0.0.1这个ip](http://www.baidu.com,就会自动被映射到127.0.0.1这个ip)上，这里就解决了本地开的时候，ajax请求因为跨域被浏览器禁止发送给服务器，

虽然本地浏览器跨域请求发送的问题解决了，但url请求时定位到了自己本机的127.0.0.1上，那本机就是作为后台服务器了，如何处理这个请求呢，这个时候就需要设置本地的nginx来做“反向代理”，nginx本地服务器启动，静态资源在nginx本地服务器上访问【ngibx不做处理】，

跨域api通过nginx的代理设置，去请求真正的后台【需要配置nginx，配置很简单的】；说白了具体的流程就是：

1. 本地开发环境的域名和api域名相同，本地浏览器发送请求，因为域名相同，所以本地发送没有跨域问题；
2. 请求发送出去结果hosts重定向，定到本机的ip【127.0.0.1】；
3. 本机下设置了nginx，静态资源请求本地的静态文件，api请求通过nginx转发请求正确的后端资源，然后返回给本机，实现了本地开发环境跨域

为什么ip用127.0.0.1这个本地ip而不是远程的静态ip，因为远程的ip的静态资源是不变的，而本地开发的时候，自己的静态资源随时会变，所以需要把hosts文件中的ip设置成本地的127.0.0.1

如果后端的api上了qa环境，还是一样，hosts中添加127.0.0.1 qa.baidu.com ,前端正常访问qa的api，nginx再对qa环境的api做定向转发

//如果前端是node自己用devserver；后端用的java，那么要请求java接口，如果java的接口域名没有配置到localhost【因为这个需要java后端去配置】，这个时候就很麻烦，要么自己本地用mock数据【这个可以整站统一设置，如果之前没有设置，那就比较麻烦，得解决调用后端跨域的接口问题】，一般就用下面的devserver的proxy

2.webpack的devServer中设置，就是设置node服务，来实现前端开发的跨域问题；应该会用到proxy字段，也是类似nginx的反向代理；例如!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!: 这个是最靠谱的，所有的前端只要拉到代码，就可以统一通过webpack实现具体某个接口的跨域，本质上就是nginx的反向代理

devServer:{//比如我想要把本地的locahost下的 locahost/getData这个接口转换成百度的<https://www.baidu.com/getData>这个接口，那么下面的pathRewrite字段就不需要，如果我想把本地的locahost/getData改成百度的<https://www.baidu.com/>接口，那么就需要在pathRewrite中设置“^/ getData”:””

proxy:{

“/getData”:{//告诉dev-server,我只对/getData这个接口做反向代理

target:<https://www.baidu.com/>,//目标接口域名

secure:true,//https的时候才用到

changeOrigin:true,//是否跨域

pathRewrite:{//重写路径

“^/ getData”:””

}

}

}

}

3.通过fiddler的正向代理，设置对应的规则，让浏览器发送出去的请求，比如点击一个js请求或者api请求，选择autoResponder，可以设置对应的匹配规则，把本地发送的ajax请求【本地发送ajax请求前缀域名还是localhost】，在匹配规则中改成统一的“后端api”的域名；

方法1：在Rulder Editor中添加一条规则，上面的是regex:http://cgclubs.com/ajax/search\?([\d\D]\*) 下面的是 regex:http://abc.com/ajax/search\?$1 ,这样就完成了把浏览器发送的cgclubs.com/ajax/search统一转化成abc.com/ajax/search类型的请求，实现跨域！！！！！！！！！！！！！！！！！

方法2：<https://www.jianshu.com/p/73530503486d>，这个最简单，直接返回头设置Access-Control-Allow-Origin：\*的返回头，“欺骗”浏览器达到跨域调试接口的效果。；当然也可以在浏览的快捷方式的后缀中添加允许跨域信息返回的问题；

当然fiddler也可以简单地把浏览器的某个请求替代成本地的某个具体的静态文件，这个是静态文件的替换，上面的是跨域api的替换。

Fiddler中的Tools的Host就是和修改hosts文件夹有相同的功能，但是fiddler不需要权限，而修改本地hosts文件夹需要权限，所以统一在fiddler中修改即可