Charles使用指导

张恒一

2018/1/5

~~*2017/7/25*~~

版本:CharlesV3.10/V4.0.2

1.原理

*Charles*是一个网络代理，可以对*HTTP*请求/响应进行监听（包括*SOCKETS*）。如图*1*所示，*Charles*可以对你发送和接收的数据进行记录和展示，然后转发。

图*1 Charles*代理原理

2.对未启用*CA*证书验证的*Https*请求进行抓包分析

安装*Charles*代理根证书，路径*Charles->Help->SSL Proxying*，有几个选项：

Install Charles Root Certificate in iOS Simulators 安装到模拟器，重启模拟器生效（*charles*需在模拟器前启动）

Install Charles Root Certificate 安装在电脑端，在钥匙串里，找到证书，设置*Always Trust*

Save Charles Root Certificate 可把证书保存在本地，通过邮件发送，移动设备下载安装，比如在*iPhone*上，在*Setting->General->Profiles & Device Management*找到*Charles*根证书信任即可

其它方式：

使用浏览器直接访问*http://www.charlesproxy.com/getssl/*也可安装

或者复制*chls.pro/ssl*到手机浏览器也可安装（前提是手机连接到*charles*代理）

开启*SSL*代理

*Charles->Proxy->SSL Proxy Setting*在*SSL Proxying*选项下面，选中开启*SSL*代理

添加*Location*，如*api.4001113900.com:443*（其中*443*为*https*默认端口）点击确认即可

开启*SSL*代理前的抓包如图*2*所示，可以看出请求及响应是乱码的。而开启*SSL*代理的抓包如图*3*所示，可以看出请求及响应都是明文的（其它端口可能会有问题）。

在*iOS11*及以上系统中，对*charles*根证书默认不信任。在关于手机->证书信任设置中，把*Charles Proxy*选项打开即可。

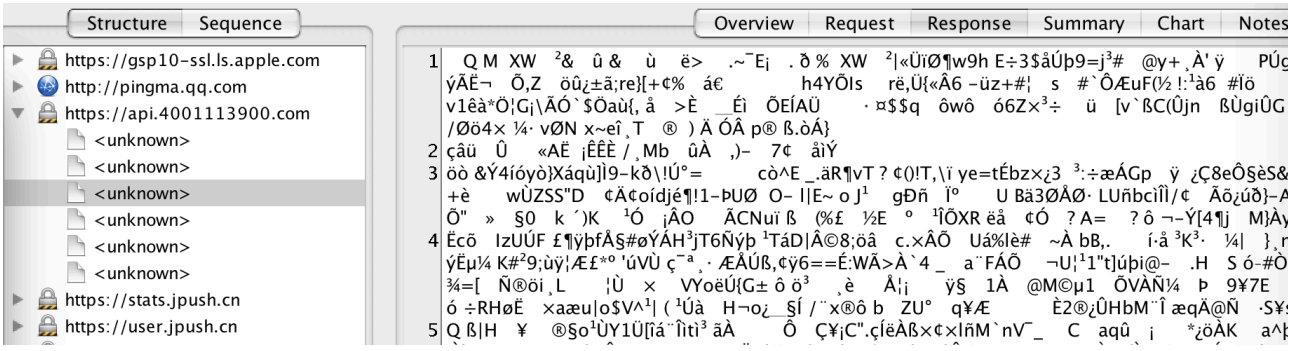

图2 关闭SSL代理的请求及响应抓包截图

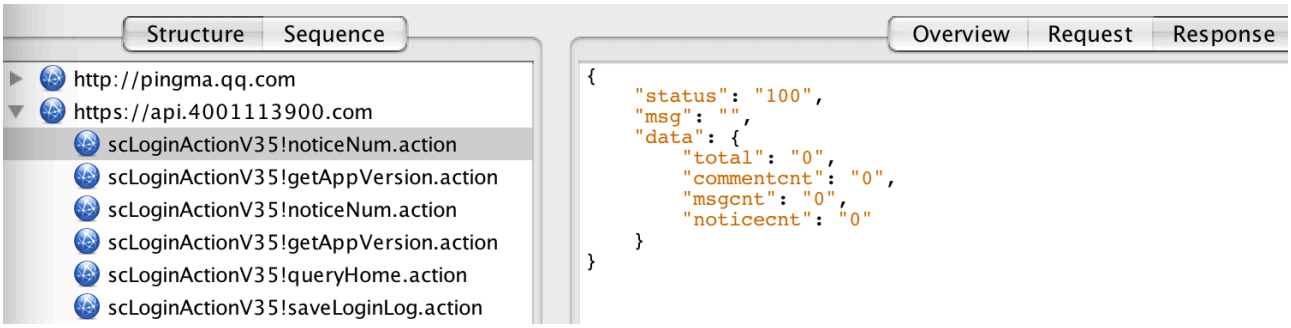

图3 开启SSL代理的请求及响应抓包截图

注：Charles只能对安装了Charles根证书的移动设备进行https抓包分析

3.对启用CA证书验证的Https请求进行抓包分析

（注：该CA证书由服务器后台开发人员提供，打包入APP端，在发起网络请求的时候进行验证，把本地的证书与服务器的证书进行比对，如果不成功，则不会进行网络请求。）

如果不进行处理，发现Charles不能对启用CA证书验证的Https请求进行抓包，如图4所示。由截图可以看出，请求没有发出，说明证书的验证失败了。

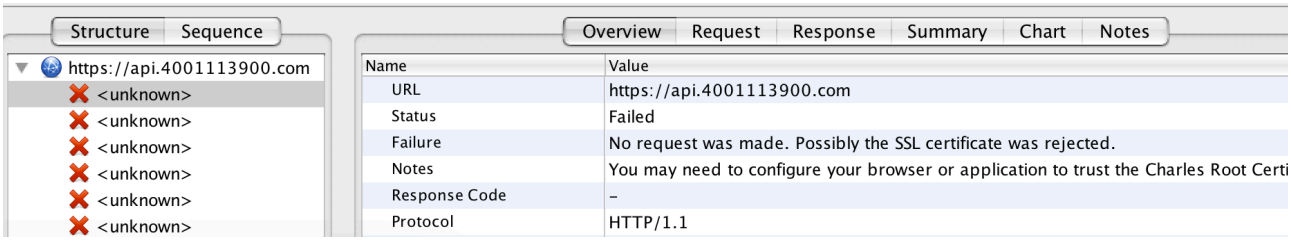

图4 Charles对启用CA证书验证的Https请求抓包截图（未处理）

如何对启用CA证书验证的HTTPS请求进行抓包分析呢?

在第2章节的基础上，再在Charles上配置一个根证书即可。该证书的格式是PKCS#12（.pfx），需向后台开发人员索取，如star.4001113900.com.pfx。打开Charles->Proxy->SSL Proxy Setting对话框，在Root Certificate选项下面，选中Use a Custom Root certificate，选择刚才的pfx文件，点击确定，会出现一个输入证书对应密码的对话框。该证书的密码也需要向后台索取。重启Charles，再次输入证书密码即生效。然后抓包分析就会变成正常的了，如图3所示。(注：此证书和密码需妥善保管，避免泄露。)

如果是自定义的CA证书，可用下面的命令生成PKCS#12证书：

openssl pkcs12 -export -out ca_cert.pfx -inkey private/ca_key.pem -in certs/ca_cert.pem

其中ca_key.pem为私钥，安全起见，后台人员不会提供。所以直接索取PKCS#12文件与密码即可。

在每次重启Charles的时候，都需要输入根证书的密码，这点比较烦。暂无解决办法。

可参考http://codeblog.shape.dk/，该作者叙述了原理，如何下次启动charles时，不再次输入密码。有三步：

生成keystore文件

keytool -v -importkeystore -srckeystore star.4001113900.com.pfx -srcstoretype PKCS12 -srcstorepass <your_key> -destkeystore keystore -deststoretype JKS -deststorepass Q6uKCvhD6AmtSNn7rAGxrN8pv9t93

keytool -alias 1 -keypasswd -new Q6uKCvhD6AmtSNn7rAGxrN8pv9t93 -keystore keystore -storepass Q6uKCvhD6AmtSNn7rAGxrN8pv9t93 -keypass <your_key>

替换charles.jar文件中默认的keystore

jar vfu /Applications/Charles.app/Contents/Java/charles.jar keystore

但按其步骤试了一下，没有效果。

Note that Charles asks for the certificate's password during every startup ... but if you use Charles's builtin certificate, it won't ask you for a password.

在*Charles4.0.2*版本中，在*Charles->Proxy->SSL Proxy Setting*对话框，*Root Certificate*选项下面，需要选择*p12*格式的证书。把*pfx*证书的后缀直接改成*p12*即可，输入密码并选择记住，下次打开时就不需要再输入密码了。

*4.*对网络请求进行限制

路径*Charles->Proxy->Throttle Settings*

开启限制，可对指定的*host*进行限制，如图*5*所示。包括上传下载的带宽，延迟，利用率，数据包大小*(MTU)*等。*Charles->Proxy->Start Throttle*选项可开始限制，*Stop Throttle*可关闭限制。

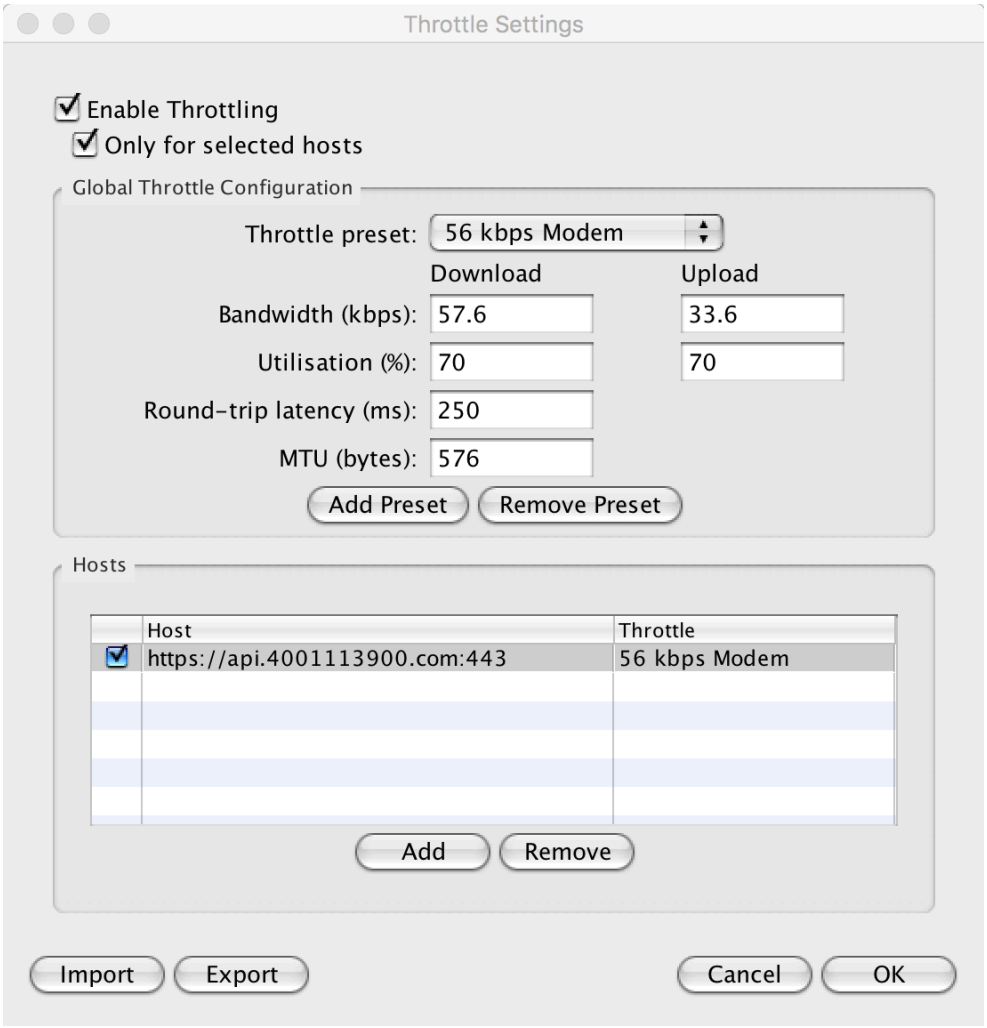

图*5* 对网络请求进行限制

*5.*对网络请求的响应进行篡改

在*Charles*界面的左侧结构选项下面，选择一个接口请求，右键选择*Save Response*（保存响应）选项，如图*6*所示。保存到本地之后，可以随意更改，注意格式要符合要求，不然*APP*解析不

了。然后在选项刚才的右键，选择*Map Local*选项，如图*7*所示。在*Map To*下的*Local Path*输入框中，填写刚才保存的响应路径。下次对该接口进行请求的时候，会返回本地的响应给客户端。

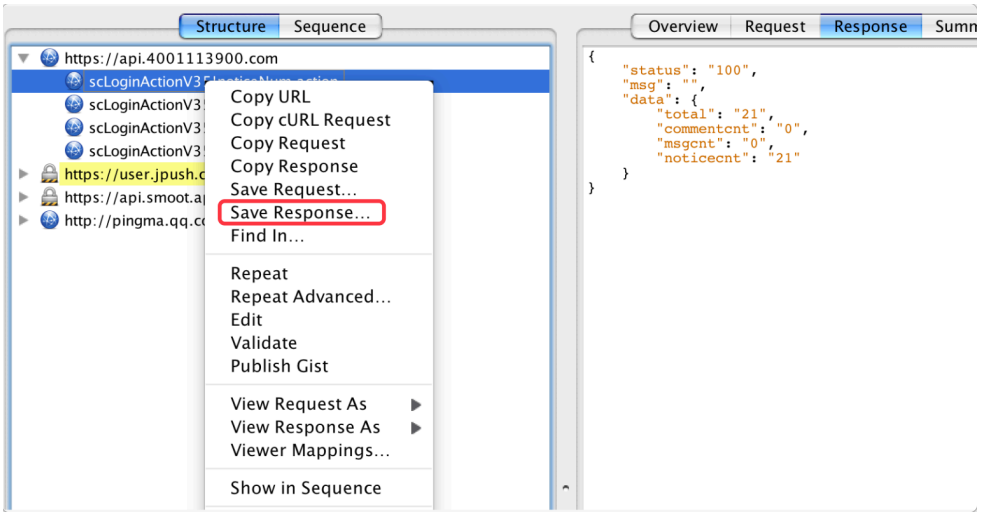

图*6* 请求右键选项截图

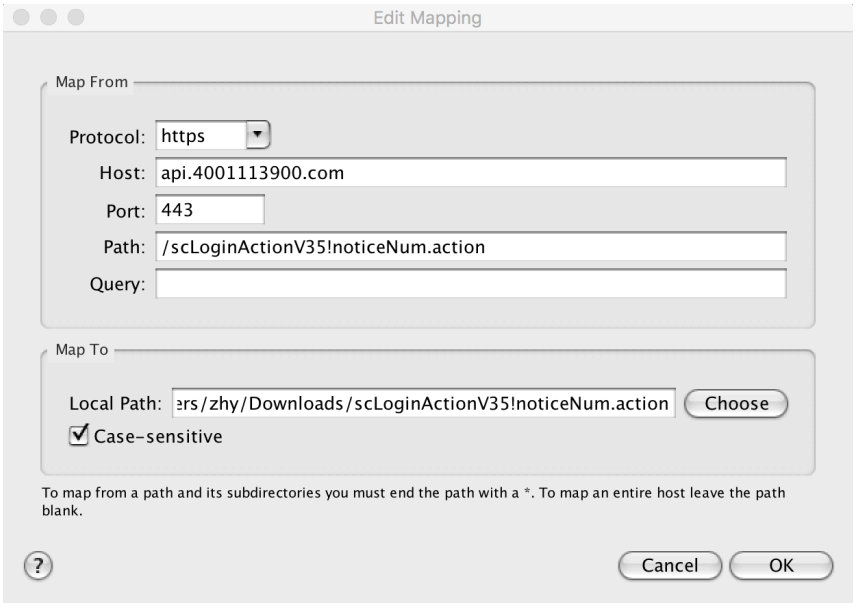

图*7 Edit Mapping*截图

修改网络请求的响应，便于*APP*进行调试，不用服务器端造数据，改数据等动作，大大提高了效率。但具体的业务逻辑还需与后台进行对接。

*Charles*篡改后，*APP*解析可能会出现不支持格式的情况，如*NSLocalizedDescription=Request failed: unacceptable content-type: text/plain*。由于*Charles*的文件格式是*text/plain*，所以需要对请求的*acceptableContentTypes*进行设置，把*text/plain*添加进去即可。

在*Charles->Tools->Map Local*中，可以把读取本地的请求给取消掉，正常的进行请求/响应转发。所有的*map local*列表都展示在图*8*中。其中*"Enable Map Local"*可用来开启或关闭本地重定向。在列表中，只有被选中的条目才会起作用。

重定向使用的保存*json*串的文本文档时可能不方便阅读，这时可借用*http://www.bejson.com*，把*json*串校验一下，格式会变得容易浏览，修改起来也比较容易。

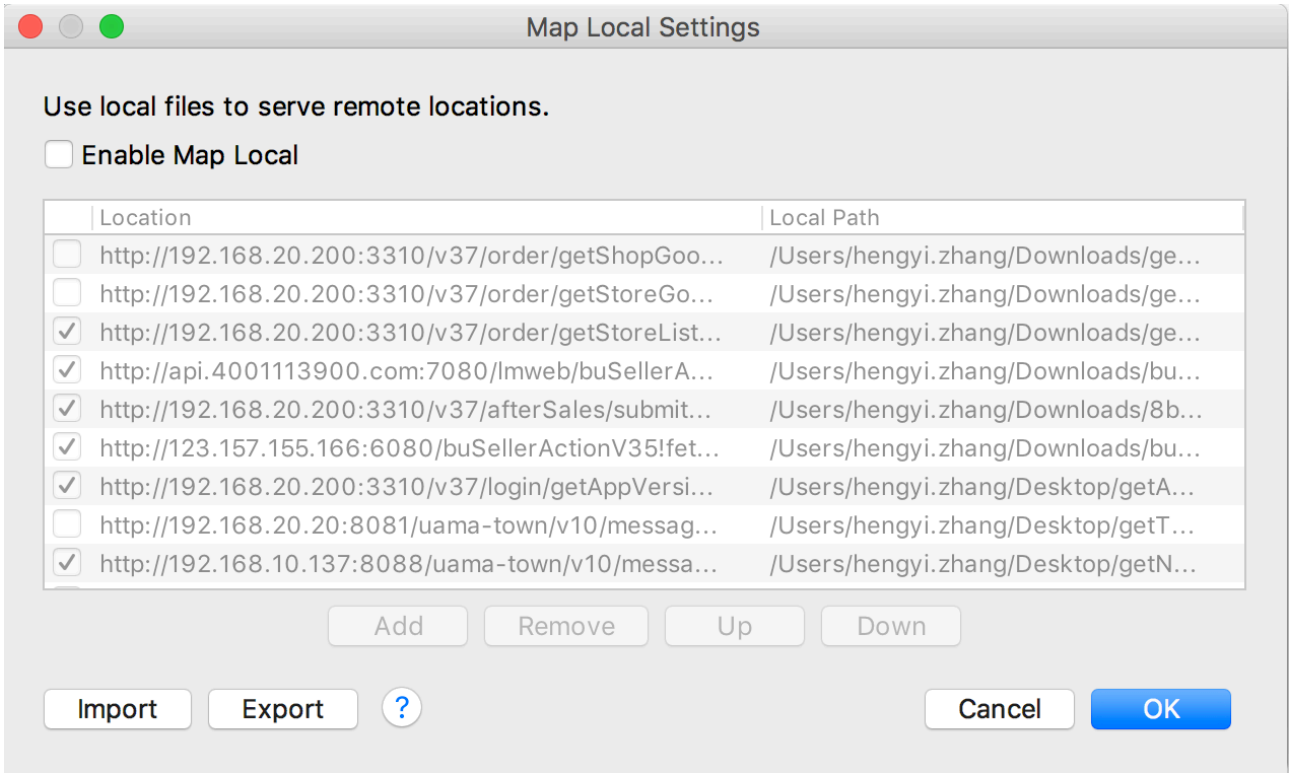

图*8* 重定向到本地设置

*6.*对请求的服务器地址进行重定向

打开*Charles->Tools->Map Remote*，如图*9*所示。

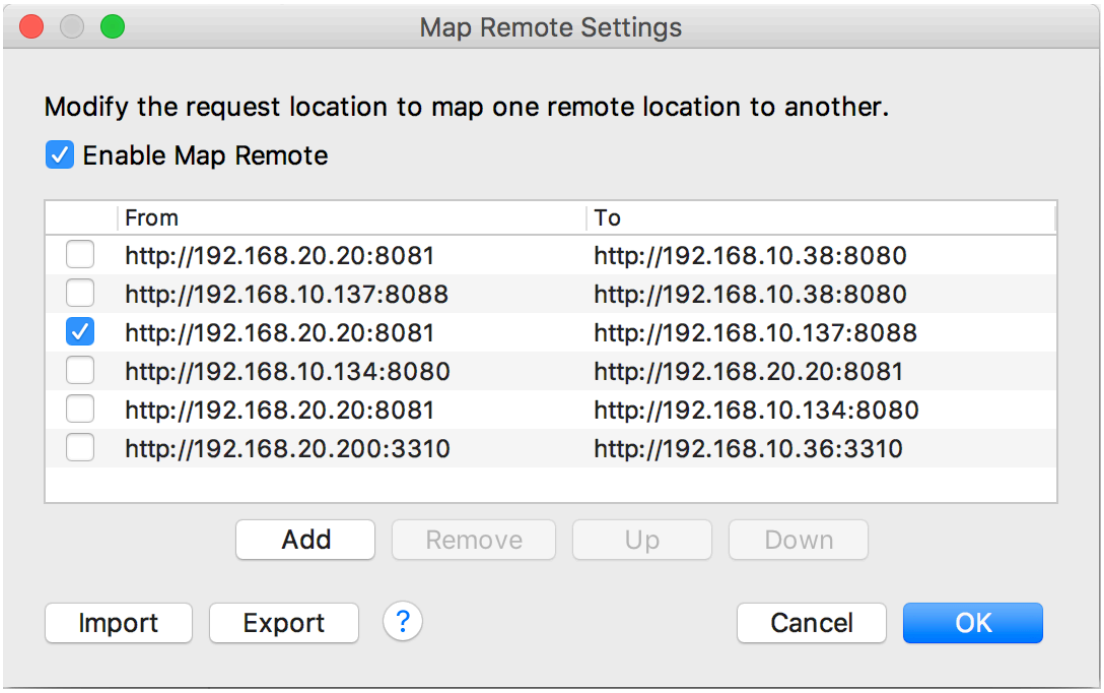

图9 重定向设置窗口截图

点击添加按钮，如图*10*所示，可添加一个重定向设置。从截图上可以看出，其可以重定向协议*protocol*、服务器*host*、端口*port*、接口路径*path*到查询条件*query*。在*host*输入框里面，输入如*“http://192.168.20.200:3310/v37/order/getFetchHomeAllModelItemList?menuData=1”*，然后点击*tab*键，会自动分解该输入至相应的输入框中。*Map from*为需要定向的源位置，*Map to*为需要定向的目标位置。

此功能的作用是，比如当我拿到一个已经打好的御姐包，我需要查看在开发环境或者正式环境的表现。使用此功能，可以很方便的进行切换，而不用重新打包。

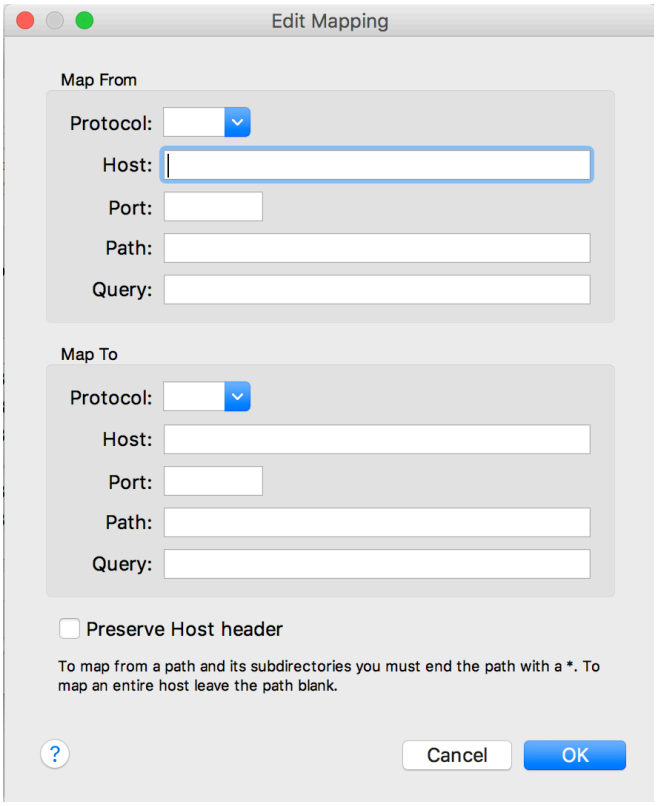

图*10* 编辑重定向

*7.*请求记录设置

当打开*charles*时，随便访问一个网页，或者打开其它*app*，左侧会显示许多的记录，不便于自己调试的接口的查找。找到*Charles->Proxy->Record Settings*路径，定位到*include*选项卡，如图*11*所示。可以添加需要记录的服务器、端口、请求路径、查询条件。只要有一个选中，在*charles*的主界面的左侧，就只会显示所选中*location*的记录，其它的则不会出现。

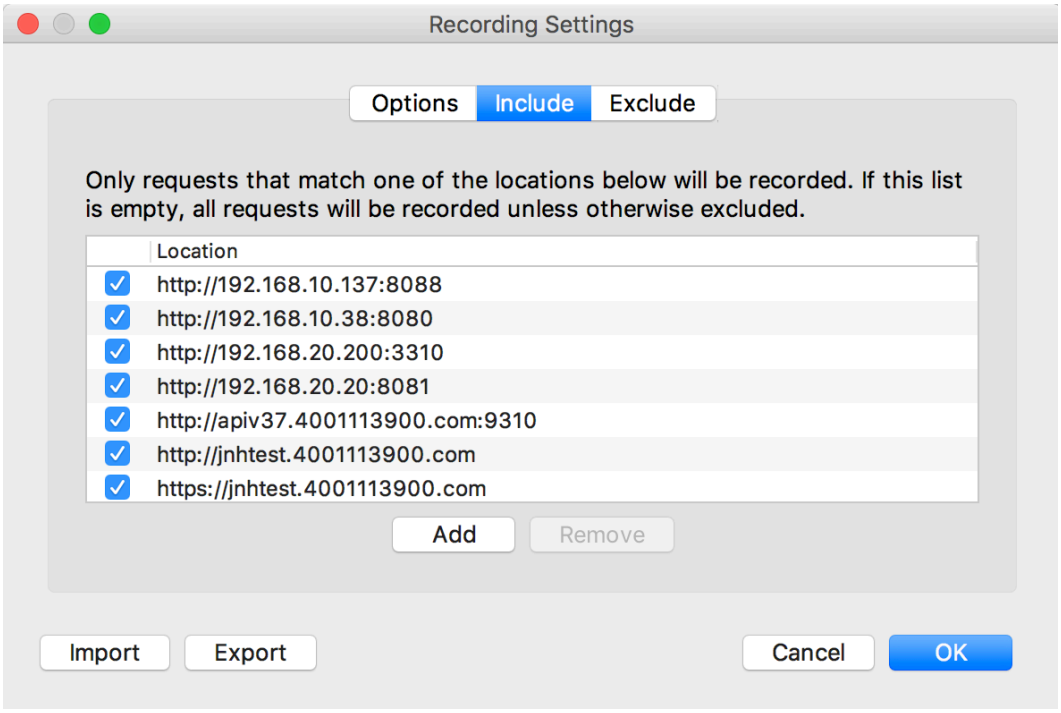

图*11* 记录设置屏幕截图

注：更多功能及使用方法，在后续开发中如果用到会补充。