方法原理：

利用SoapUI创建自定义接口，再使用抓包工具控制App请求自己接口数据，以达到模拟接口数据的效果。

使用场景：

1. 在抓包工具请求本地文件失效时，例如在Android上模拟数据时，有时会校验不通过。
2. 模拟接口返回结果比较多时，利用SoapUI控制返回数据会比较方便。

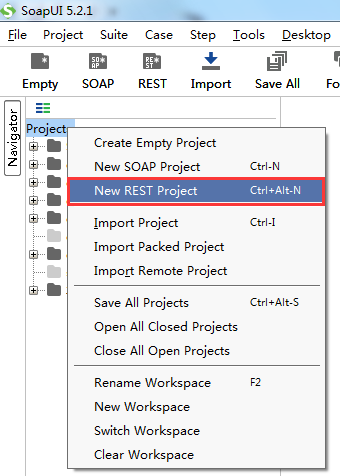
测试接口：

Xxx，已删除，自己找个吧

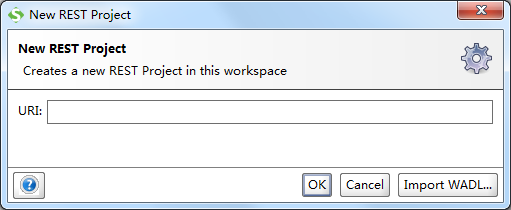
一、先用SoapUI创建接口：

1. 创建Rest Project项目

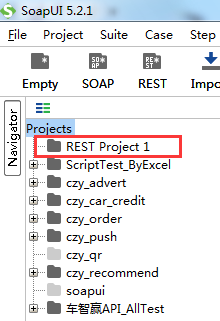
打开Soapui，左侧导航栏选中Projects右键，选择”New Rest Project”，创建一个Rest项目。



会弹出New REST Project弹窗，直接点OK：



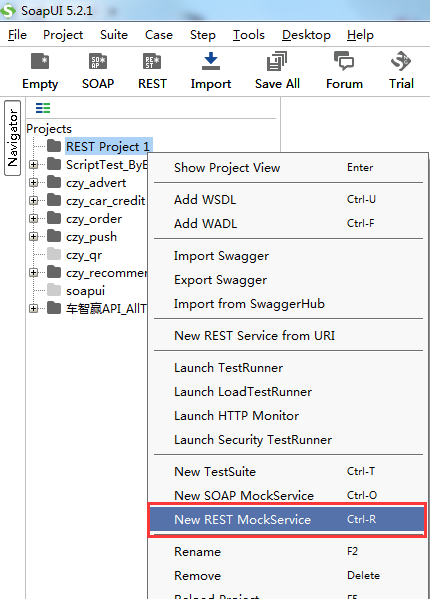
RestProject就创建好了：



1. 创建自己接口：

2.1 创建MockService：

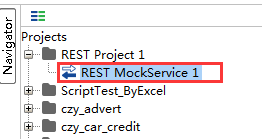
新建的Rest Project上右键，选择”New REST MockService”。



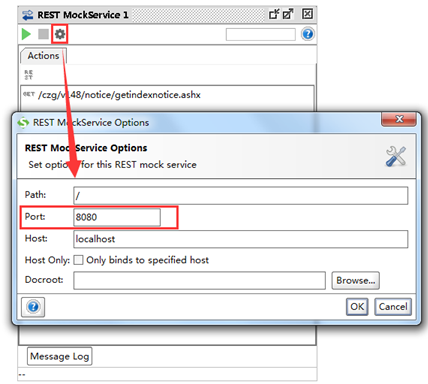
默认命名或者改名都行，点ok。



Rest MockService就创建好了：

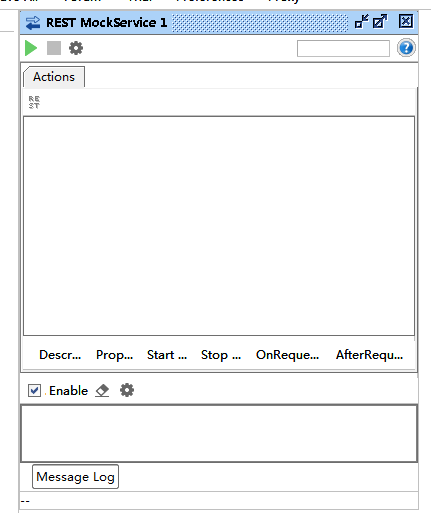


另外，Mock Service上可配置自己接口的端口号和域名，默认8080和localhost，如图：

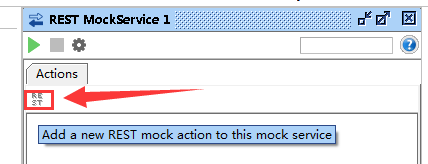


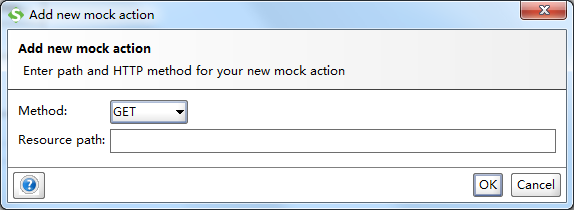
2.2 创建Mock Action（编辑自己接口的Path）：

双击新建的MockService，右侧弹出窗口：

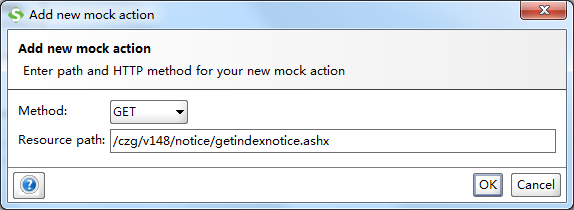


如图，点击左上的”REST”图标按钮，创建Action：

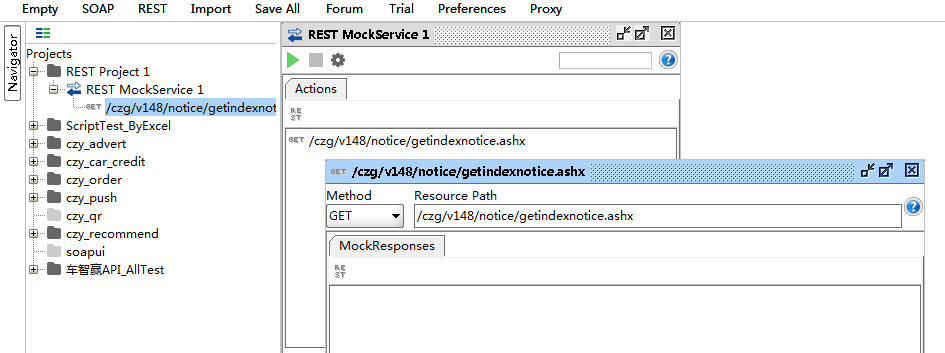




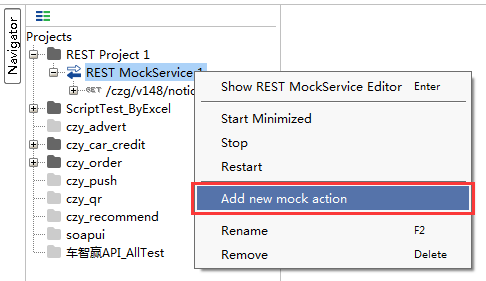
此处输入自己接口的Path路径，我一般习惯写和原接口相同的路径。



填写后，点OK，MockAction创建好了：

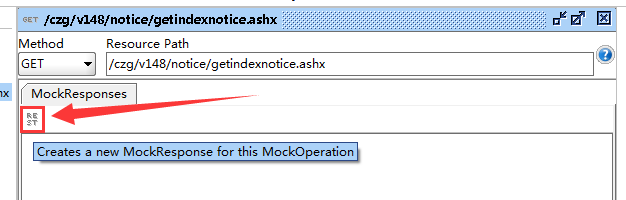


（另外，Mock Action可通过左侧导航栏，右键快速新建，如图：

）

2.3 创建MockResponse（可导航右键快速新建）：

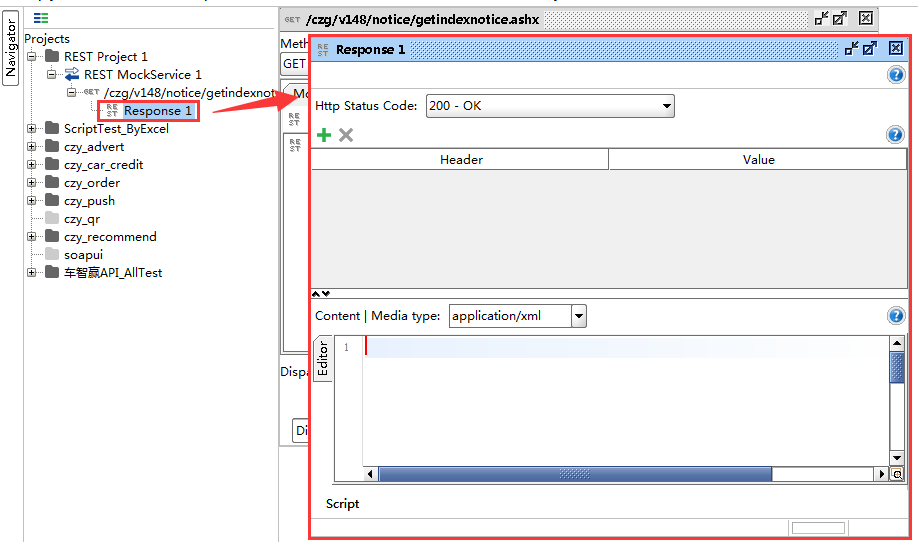
同Action，双击新建的MockAction，弹出窗口，点击左上的”REST”图标按钮：



默认命名或者改名都行，点ok。



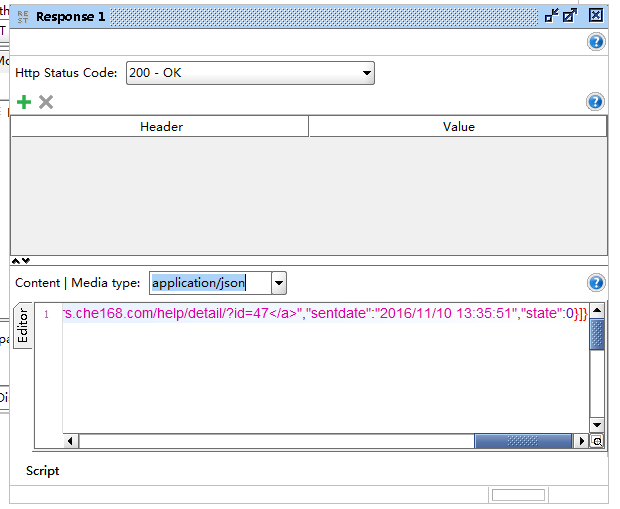
MockResponse就创建好了：



顶部是Http接口状态，上半部是Response Header部分，下半部是Response Body 部分。Response Body可在Editor输入框里编辑接口内容。

2.4 编辑Mock Response（编辑自己接口的Response Body）：

双击”Response 1”，弹出窗口，Content|Media Type选择” application/json”，然后在Editor编辑自己接口的返回数据：



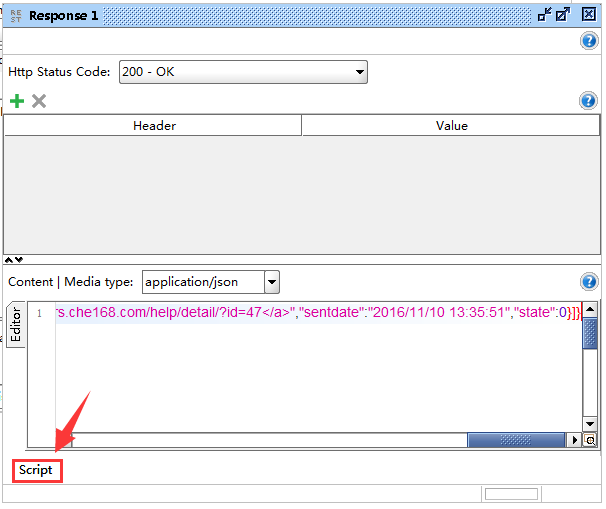
Json最好自己格式化下，开源版的SoapUI不管格式转换。

2.5 编辑Response Header：

（编辑自己接口的Response Header，内容拷贝自服务器接口的Header 信息）

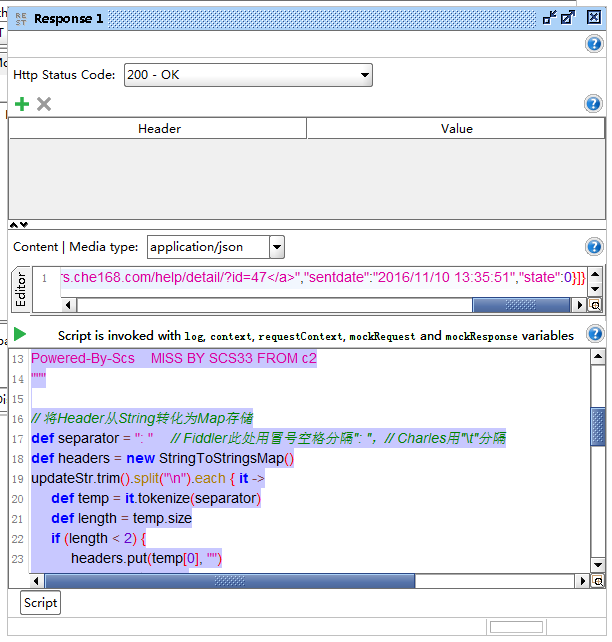
2.5.1黏贴脚本：

点击右下的”Script”按钮，打开脚本编辑器，黏贴输入脚本：



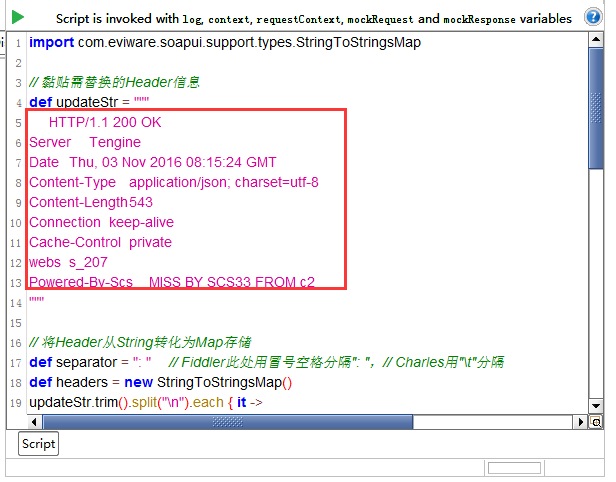
复制黏贴以下脚本内容：



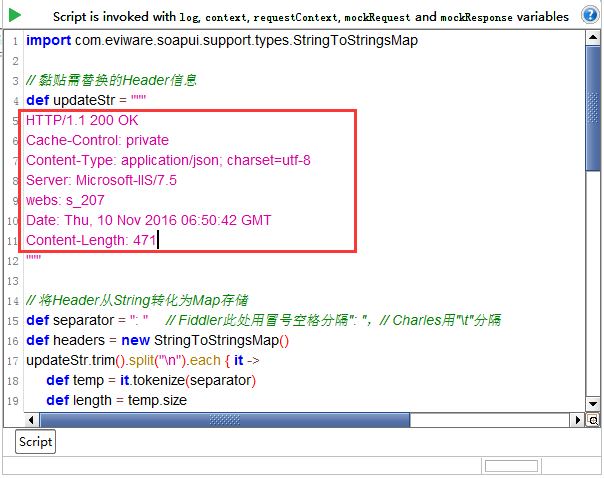


2.5.2 修改脚本里的Header信息，并运行脚本，完成替换Header的操作：

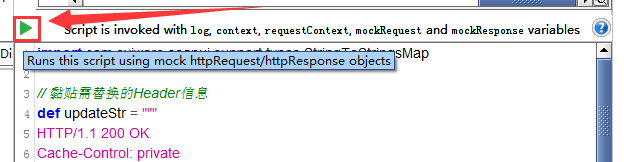
原有：



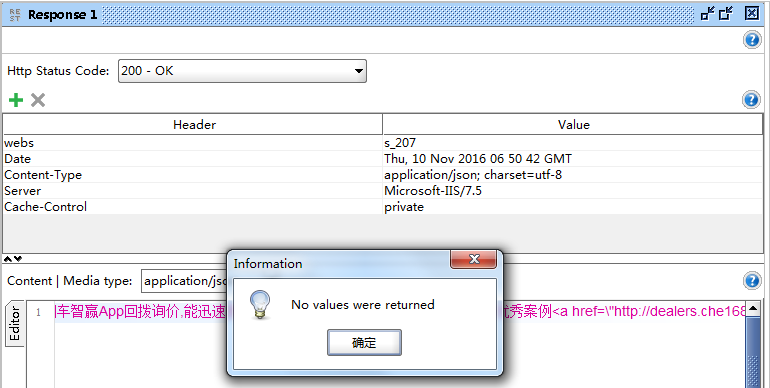
替换：



然后点击脚本输入框左上的绿色三角，运行脚本，就完成了替换：

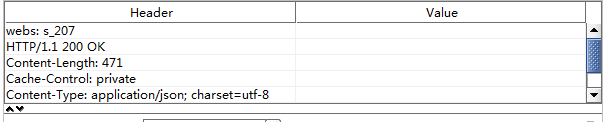


弹框，表示运行完毕。同时上面Header部分也有了内容：

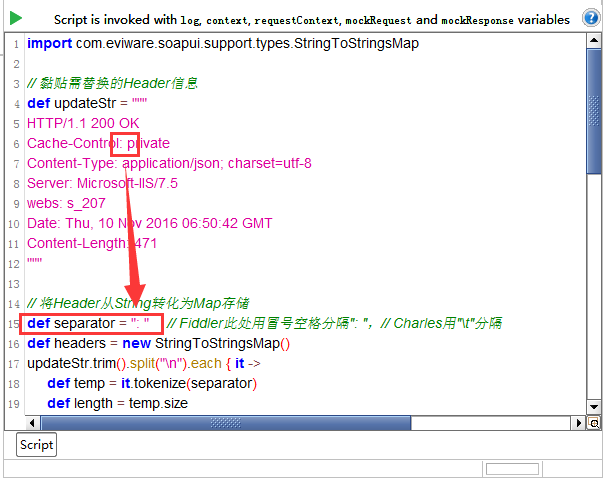


2.5.3 错误排查：

若运行脚本后，Header信息填写有误。如：



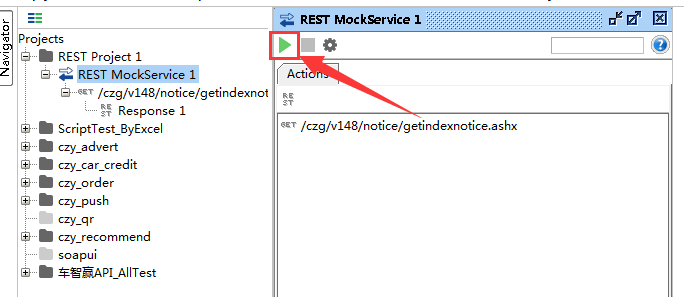
请查看替换后Header信息的Key和Value之间是什么字符。是冒号”: ”、还是空格” ”、或是Tab，然后修改脚本中separator的值。然后重新运行，应该就可输入正常。



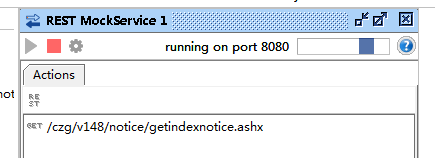
2.6、启动模拟接口：

2.6.1 启动：

双击Rest MockService，会弹出REST MockService 窗口，然后点击左上的绿色三角，即可启动接口：



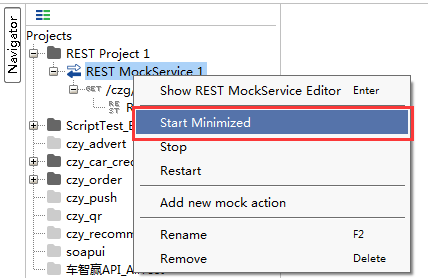
窗口右上出现端口和滚动条，代表启动成功：



之后，打开浏览器，输入127.0.0.1:8080，加上接口Path路径。即可查看模拟接口内容。

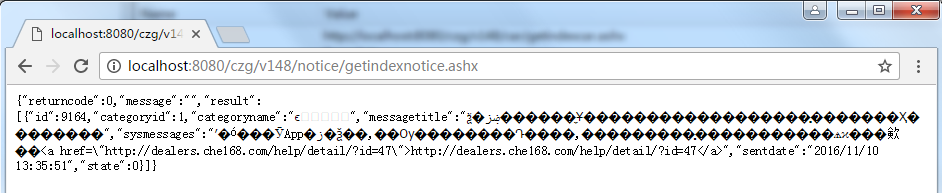


请求有结果，代表启动成功。另外，启动也可通过导航右键快速启动：

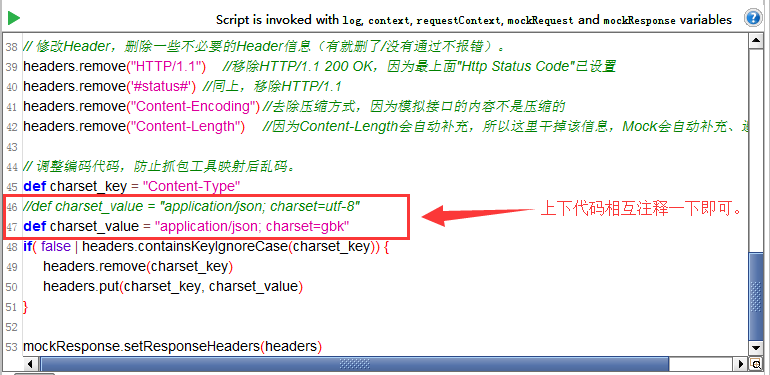


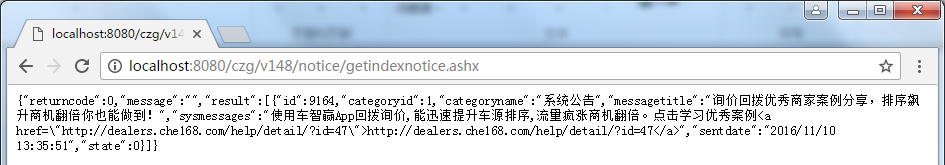
2.6.2 错误排查：

若接口返回的结果中文乱码，如图：



则请修改脚本代码中的charset\_value的值，将charset=utf-8改为charset=gbk，或者反过来。重新请求接口即可正常。如图：





二、使用抓包工具做接口重定向：

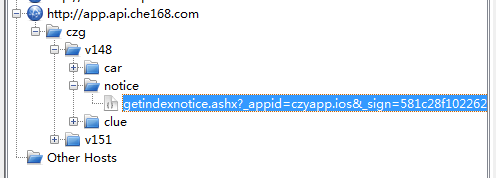
1. Charles设置接口重定向：

1.1 Charles操作，利用“Tools” –> “Map Remote” 功能，设置接口重定向：

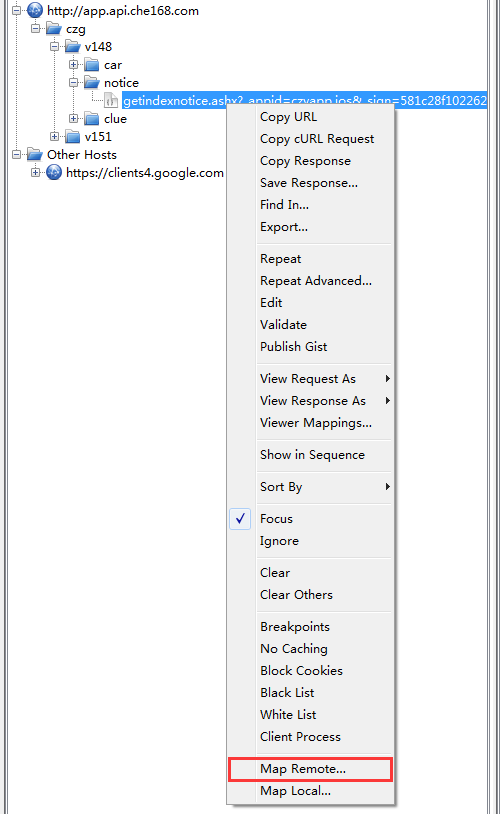
功能简单解释：

Charles 的Map 功能分Map Remote 和Map Local 两种，顾名思义，Map Remote 是将指定的网络请求重定向到另一个网址，Map Local 是将指定的网络请求重定向到本地文件。

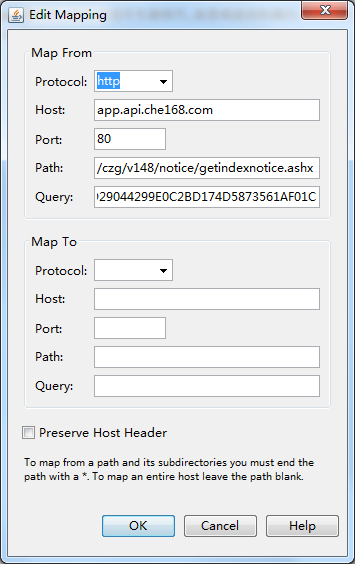
先正常抓包，触发需模拟的接口请求：



然后该接口上右键，选择最后的”Map Remote”：



会弹出”Edit Mapping”弹框：



此框里设置接口重定向关系，上面”Map From”为原接口信息，下面”Map To”为指向的接口。这里的任何字段，均可以使用\*通配符来模糊匹配。

咱们这里只重定向一个接口，因此按如下输入：

Map From：

Query的内容，全选改为\*（代表是不管给该接口传任何参数，都会跳转）。

其他不用管。

Map To：

Protocol：选Http。

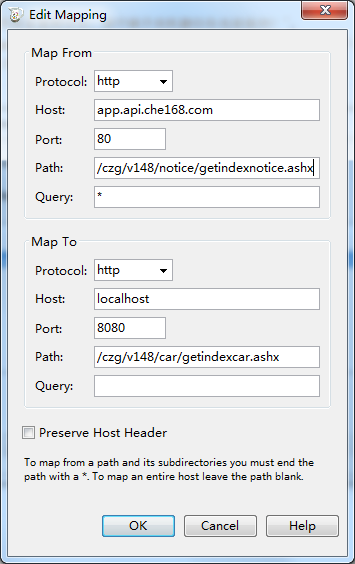
Host：输入模拟接口的域名，默认”localhost”。填写”127.0.0.1”或真实IP值也可以。

Port：输入模拟接口的端口，默认”8080”。

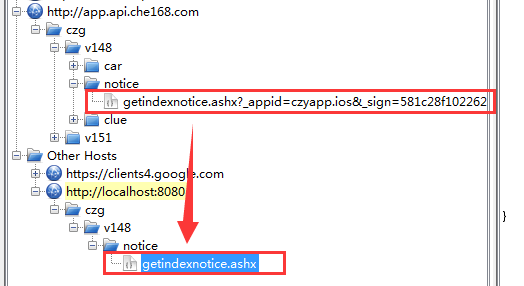
Path：输入模拟接口的路径，此处为” /czg/v148/notice/getindexnotice.ashx”

Query空着、”Preserve Host Header”不管未勾选就可以。

填写后如图：

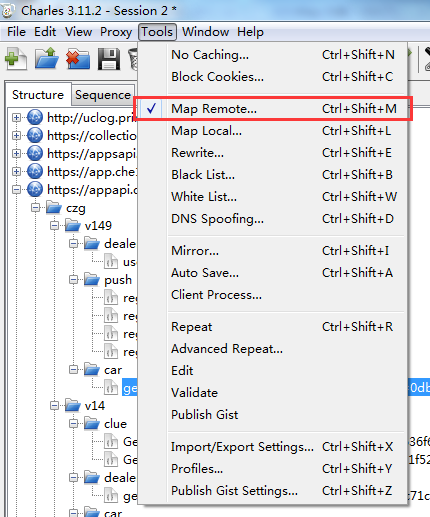


点OK保存，就开始了该接口的重定向功能了。

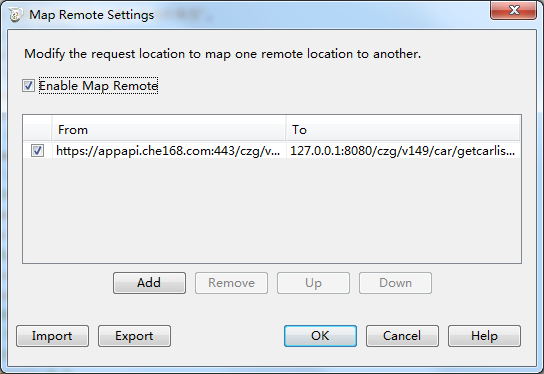


1.2 Charles操作，关闭接口重定向功能：

在Charles的菜单中，选择”Tools”-->”Map Remote…”：



会弹出”Map Remote Settings”弹框：

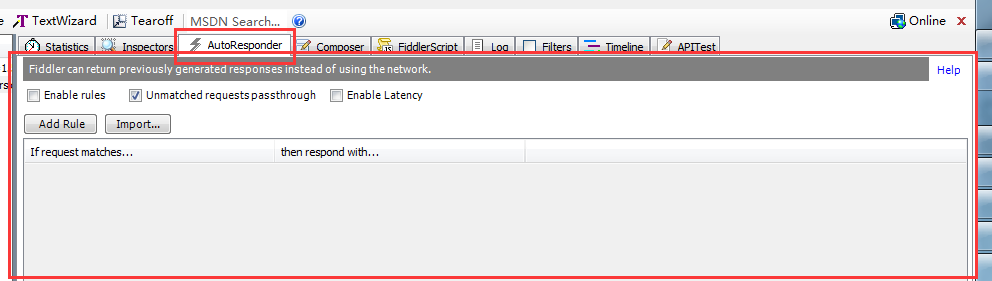


关闭接口重定向，取消”Enable Map Remote”的对勾，然后点OK保存即可。

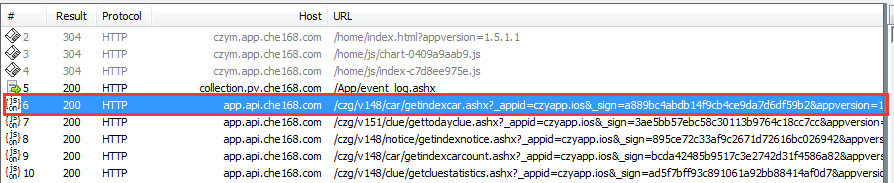
而”From…To…”列表里，可以删除重定向关系。

2、Fiddler操作，利用AutoResponder功能，完成接口重定向：

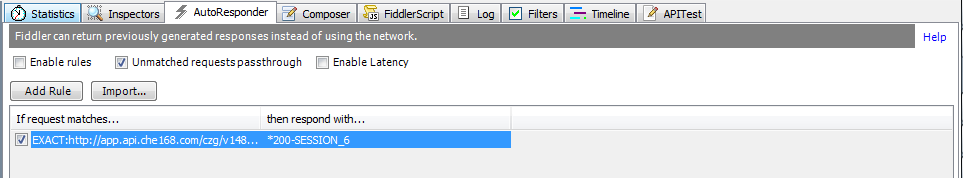
点击”AutoResponder”，进入AutoResponder窗口：

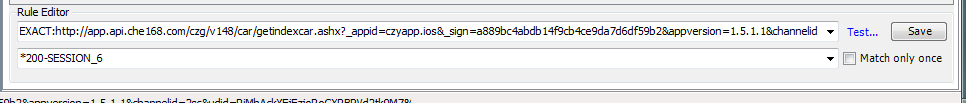


正常抓包，触发需模拟的接口请求：



左键点击按住接口，然后拖到AutoResponder窗口中，即可添加接口规则：



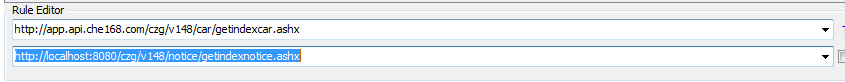


Rule Editor部分，上面是原接口信息，下面可填重定向信息，并且因为直接拖过来的规则是完全匹配规则，所以需要修改下。

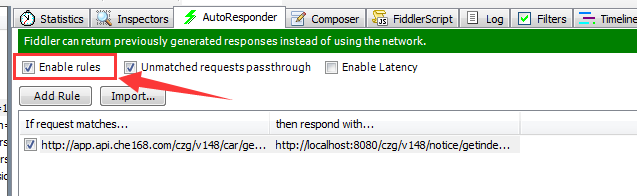
按如下修改，并右侧Save保存：

上面：将接口前面的” EXACT:”去除，然后去除后面的参数，只保留接口。

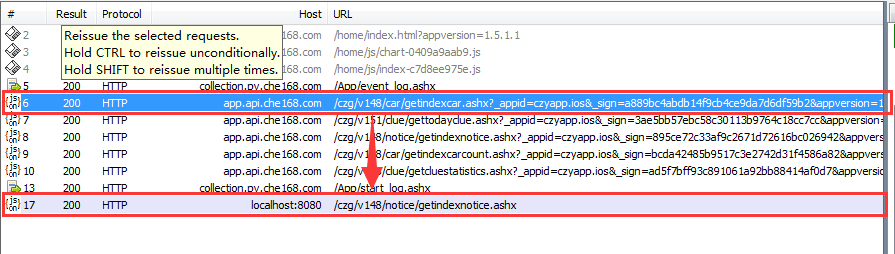
下面：填写指向后的接口<http://localhost:8080/czg/v148/notice/getindexnotice.ashx>。



然后在AutoResponder窗口左上，勾选”Enable rules”项：



就开始了该接口的重定向功能了。



3.2 Fiddler操作，关闭接口重定向功能：

在AutoResponder窗口左上，勾选”Enable rules”项

关闭接口重定向，取消勾选AutoResponder窗口的”Enable Map Remote”项，即可。

而”If request matches…then respond with…”列表里，可以删除重定向关系。

完毕。