

Charles课件

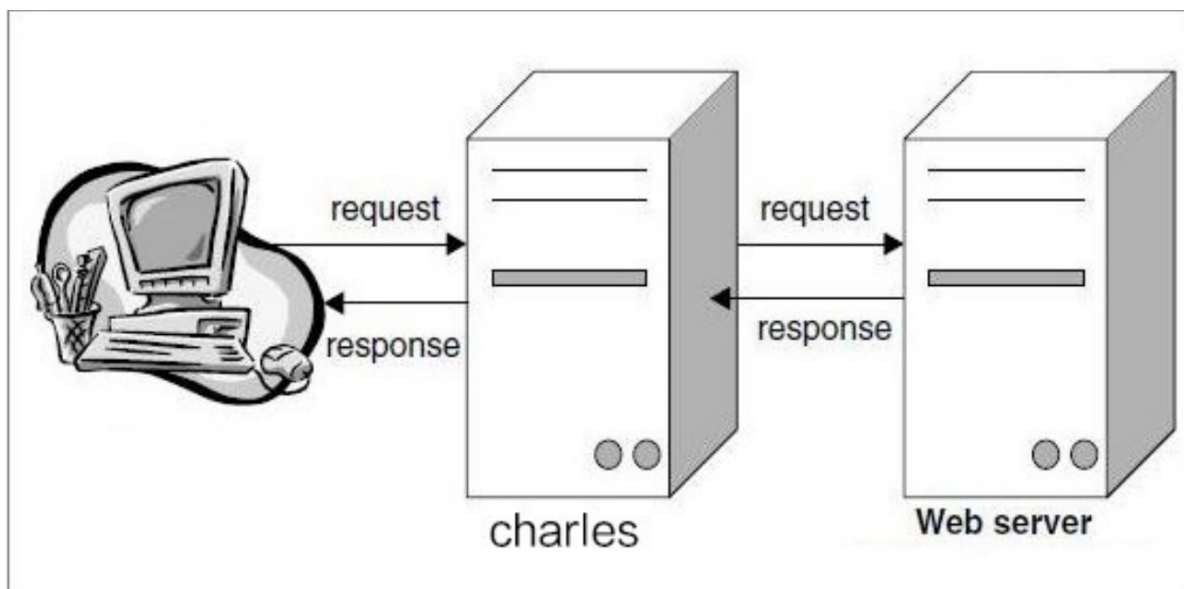
1.charles简介

charles是一款非常优秀的抓包工具，全平台支持，在mac，windows，linux上都可以使用，既可以抓取web端的包，也可以抓app端的包。

charles主要的功能包括如下几点：

- 截取 Http 和 Https 网络封包。
- 支持重发网络请求，方便后端调试。
- 支持修改网络请求参数。
- 支持网络请求的截获并动态修改。
- 支持模拟慢速网络

代理服务器的工作原理如下所示：



客户端发起请求，请求通过charles转发给服务器，服务器返回响应，响应通过charles转发给客户端。

charles所起的作用就相当于信使，把信息从A传递给B，并且把回信从B传递给A。

正因为他这个信使的工作，所以他对信息的内容了如指掌(不管是原信还是回信)，正因为如此，

charles也就可以篡改信息的内容，即篡改请求和响应。

2.安装

mac电脑参考这个连接进行安装和破解：<https://www.jianshu.com/p/82f63277d50f>

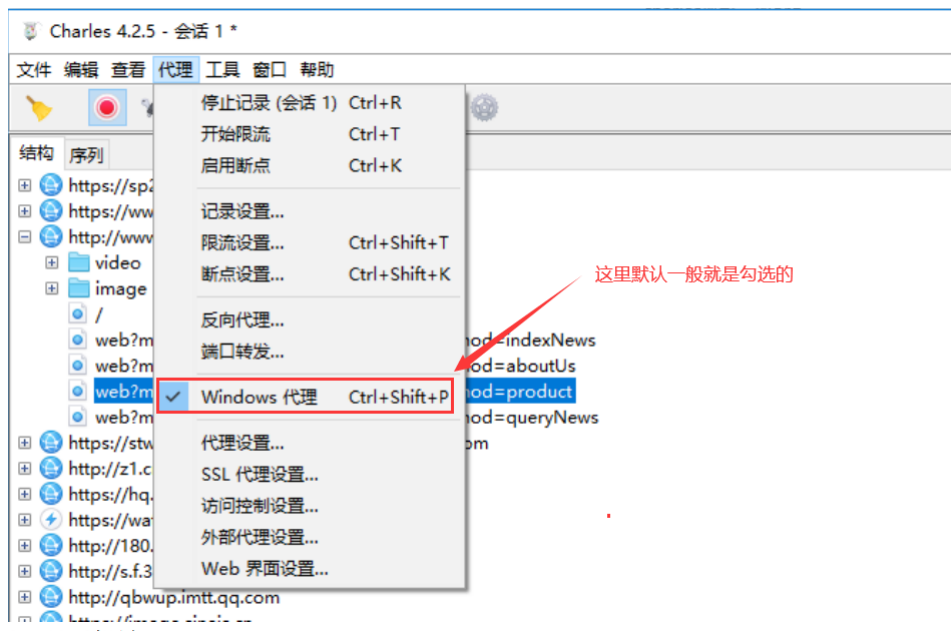
windows电脑下载网盘里的软件进行安装和破解

链接：https://pan.baidu.com/s/1AkDV5VAbVWw_uWTBW3bijw

提取码：emj4

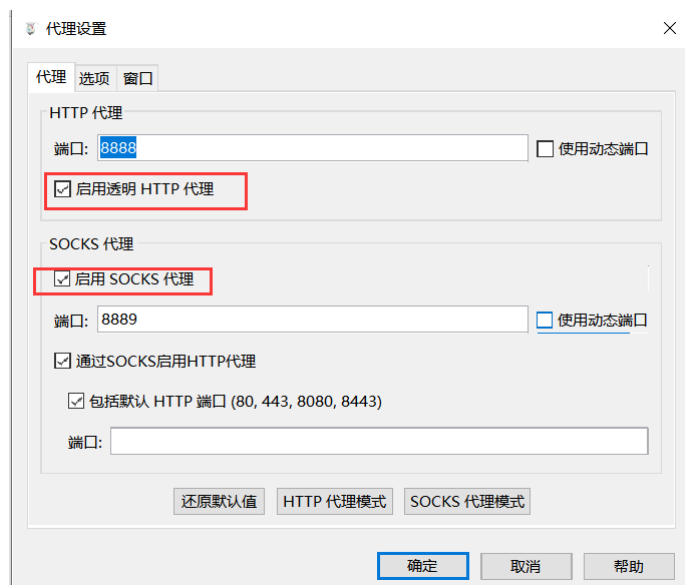
3.基本设置

- 开启代理(默认就是开启的)

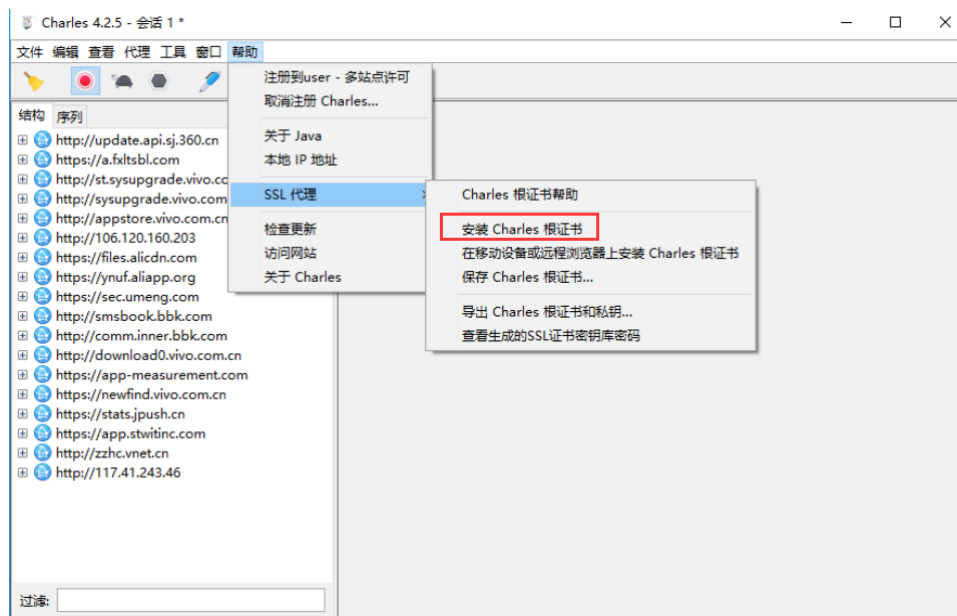


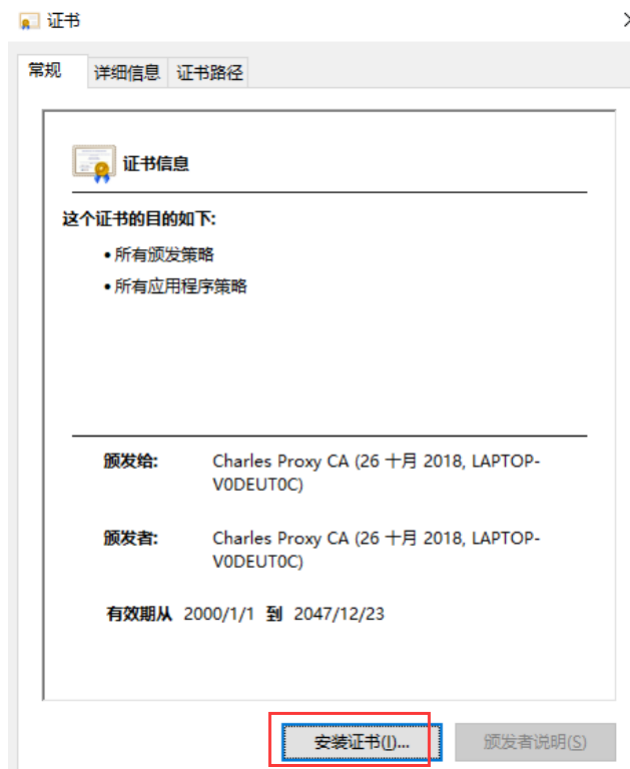
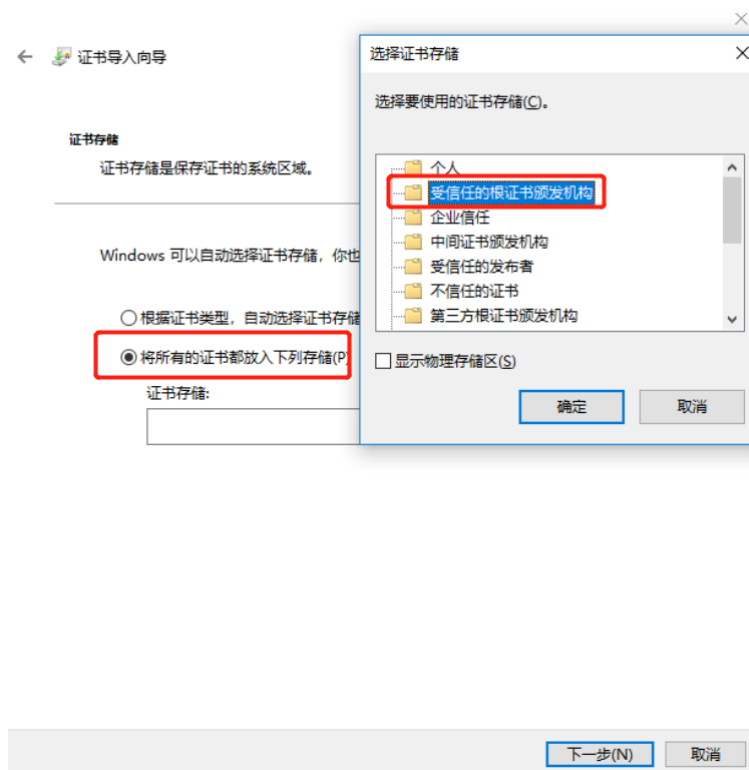
- https设置及安装证书

0. 先设置代理



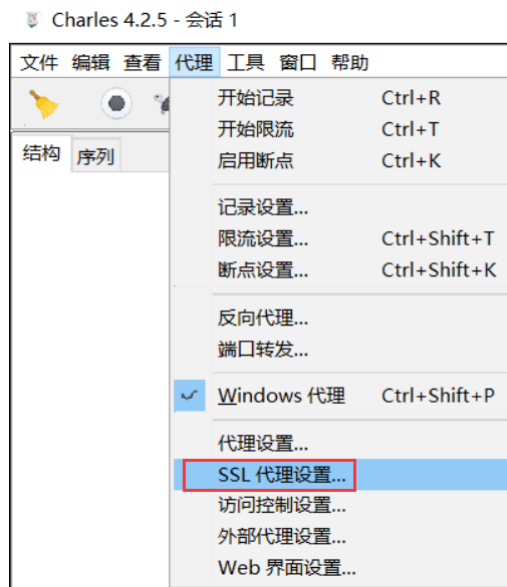
1. 安装证书



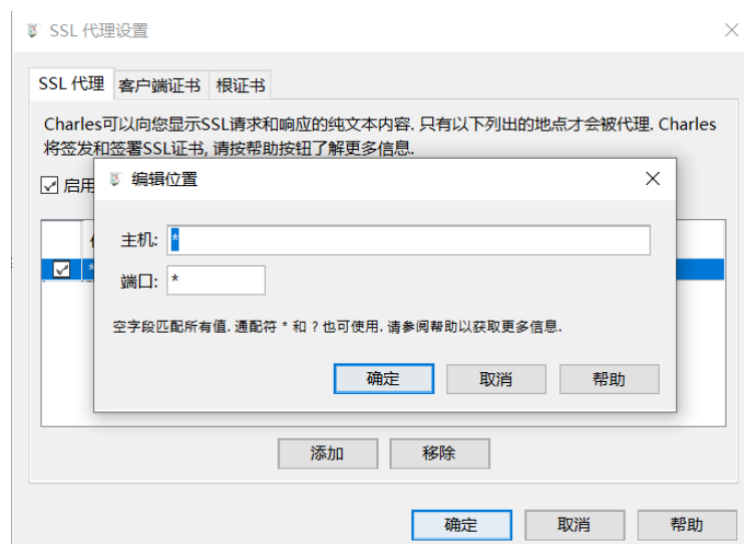


2. 设置ssl代理

单击菜单项“代理”->“SSL 代理设置”



单击【添加】，在弹出的“编辑位置”窗口中，主机、端口文本框均输入星号，如下截图所示，单击“编辑位置”窗口中的【确定】

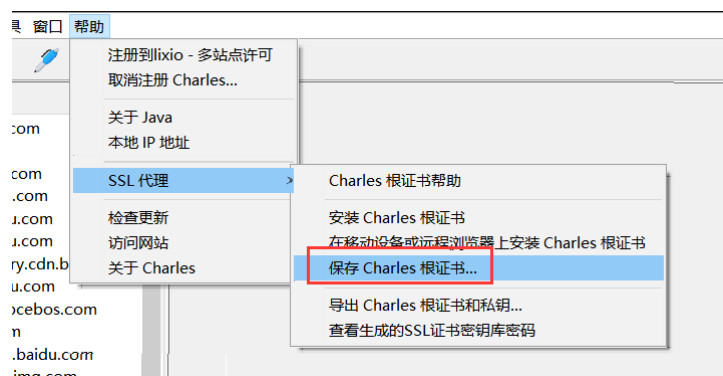


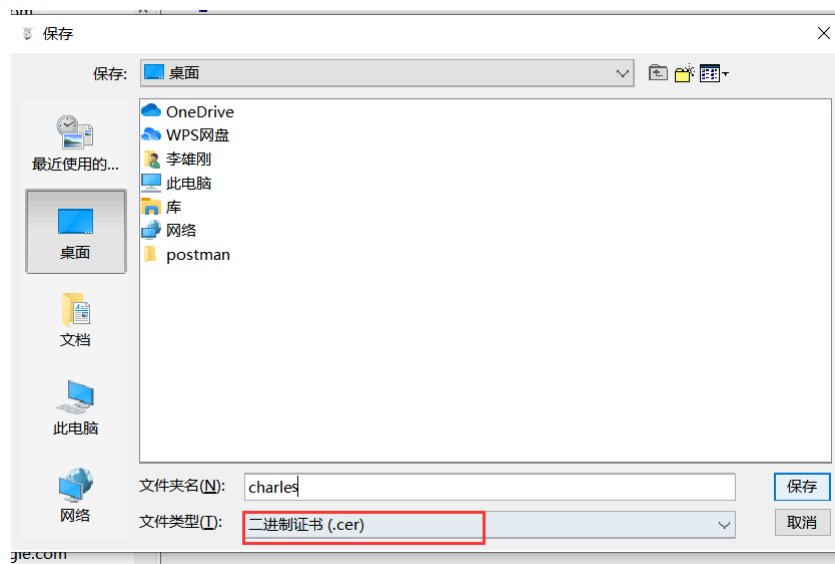
然后在“SSL 代理设置”窗口中，单击【确定】

3. chrome浏览器的单独设置

对于chrome浏览器上的https抓包需要按照如下操作进行

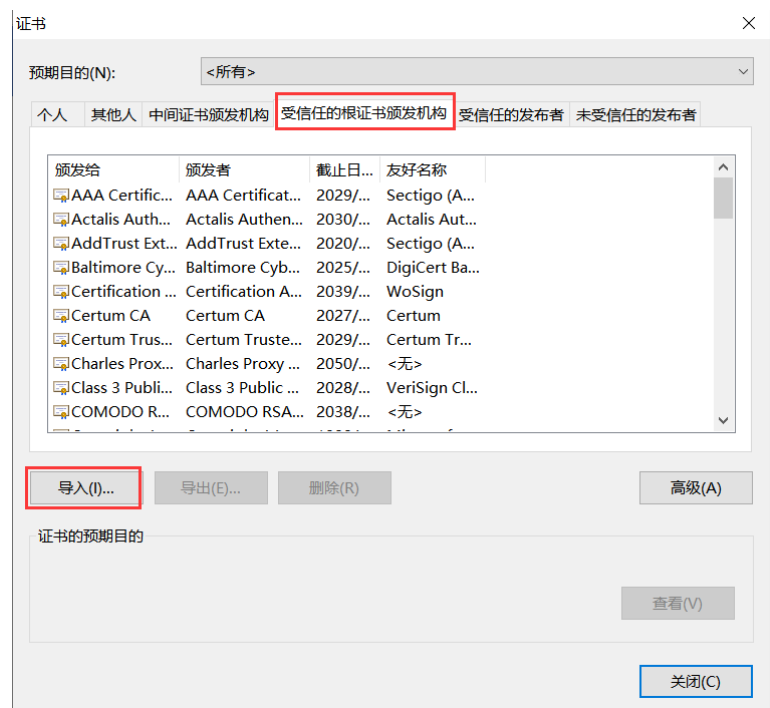
先保存charles证书，保存时选择二进制





打开chrome浏览器的设置界面，搜索证书，进入证书管理界面





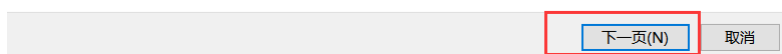
← 证书导入向导

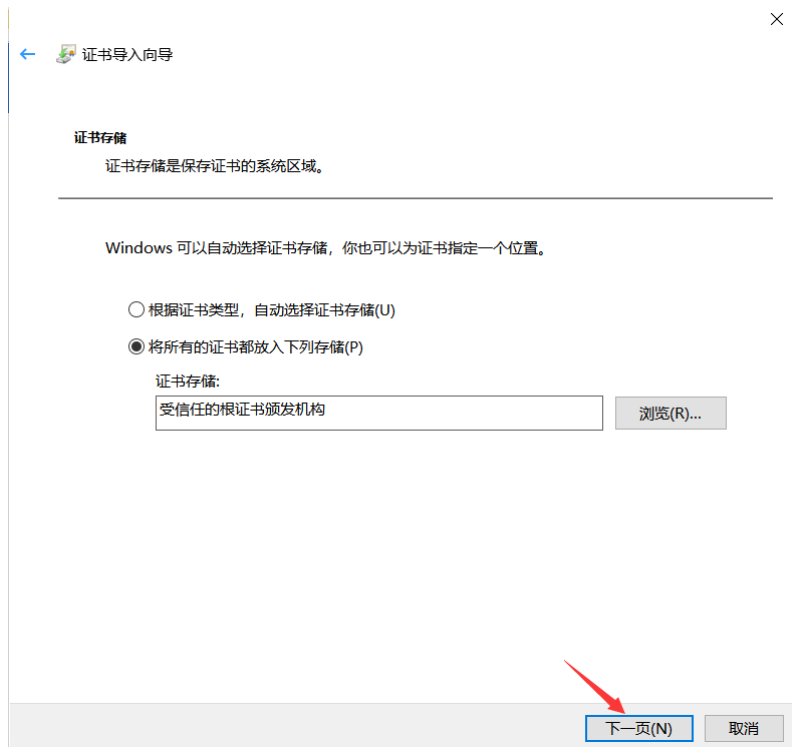
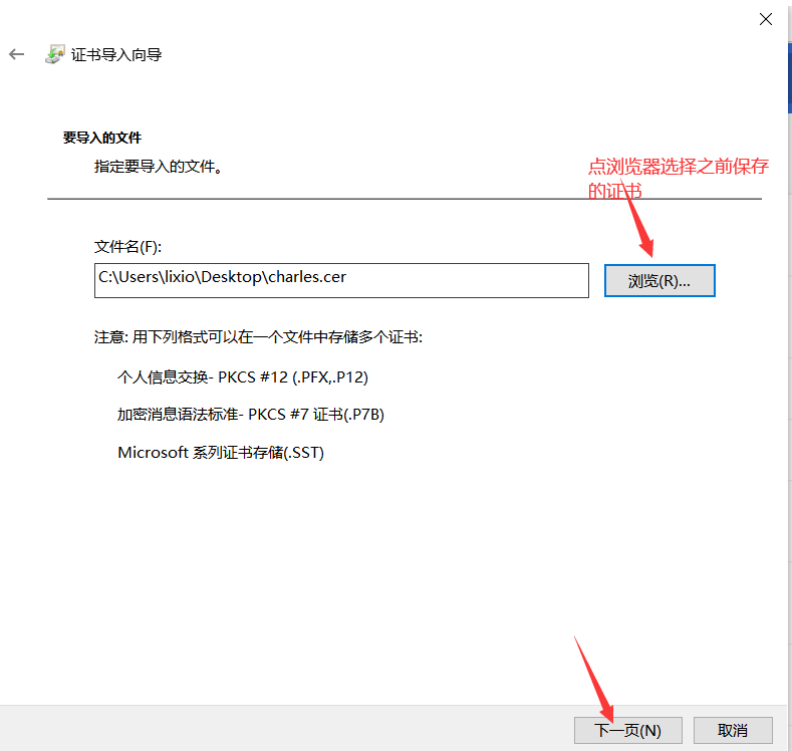
欢迎使用证书导入向导

该向导可帮助你将证书、证书信任列表和证书吊销列表从磁盘复制到证书存储。

由证书颁发机构颁发的证书是对你身份的确认，它包含用来保护数据或建立安全网络连接的信息。证书存储是保存证书的系统区域。

单击“下一步”继续。





正在完成证书导入向导

单击“完成”后将导入证书。

你已指定下列设置:

用户选定的证书存储	受信任的根证书颁发机构
内容	证书
文件名	C:\Users\lixio\Desktop\charles.cer



完成(F)

取消

4.手机端抓包配置

1. 手机连接代理设置

首先确保手机和charles所在的电脑在同一个局域网下，然后设置手机的代码连接，长按手机的wifi，打开修改网络

其中的ip是charles所在电脑的ip，端口是charles代理设置那里看到的端口号8888

第一次连接代理时，charles会弹出确认框，记得点允许



2. 手机浏览器输入chls.pro/ssl 会提示下载证书，下载并安装即可

3. ios手机还需要在设置-通用-关于本机-证书信任设置里信任证书

5.功能图标



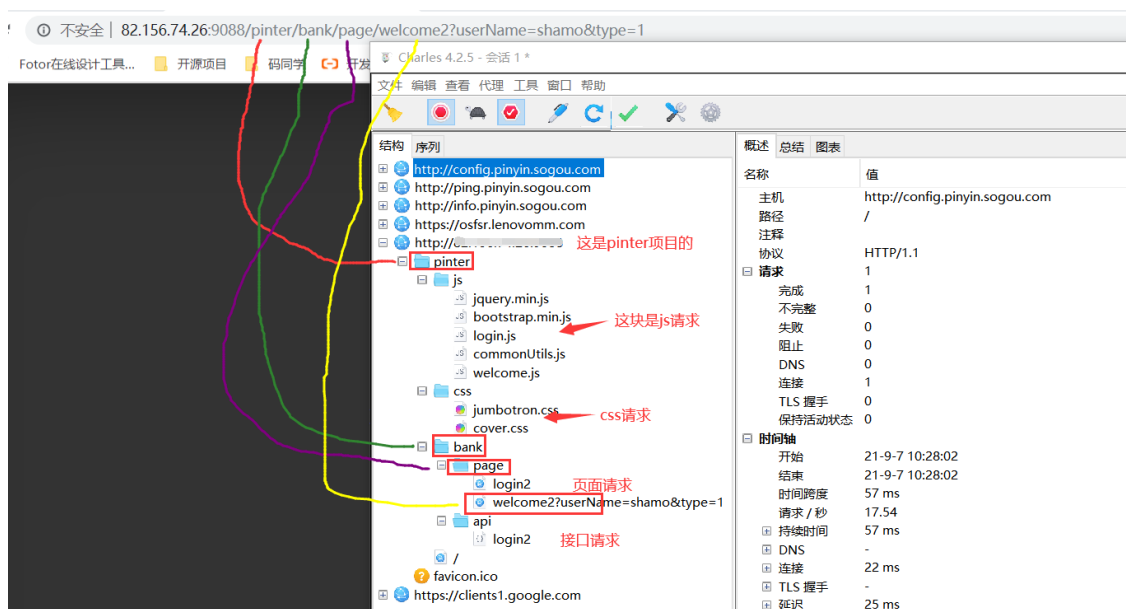
从左到右依次是

- 清除当前会话
- 停止记录
- 开始限流/停止限流
- 开始断点/停止断点
- 根据所写内容撰写新请求
- 重发选定请求
- 工具
- 设置

6.抓包视图

charles抓包的请求信息可以以两种方式展示，一种是结构化展示，一种是序列化展示

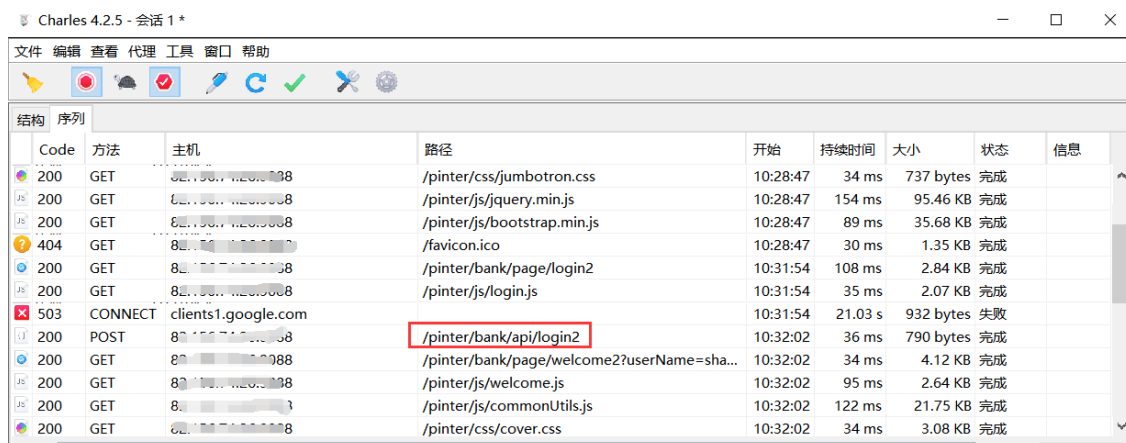
1. 结构



结构化展示方式，会以某个网址作为管理根本，然后按照各个请求的路径层级进行分类展示，如上图所示，我们最终要的纯接口

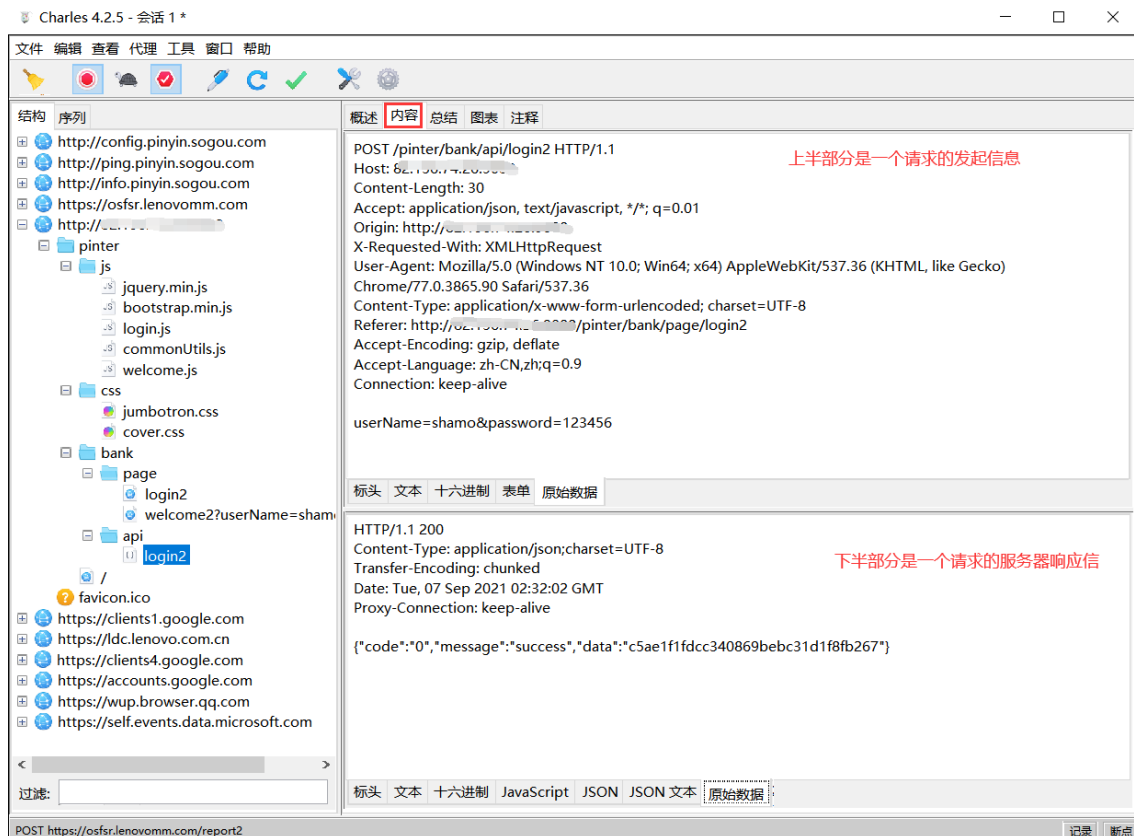
请求实际上是api下的login2

2. 序列



序列是按照抓到的请求逐个排列的，一行就是一个，每一列都代表着不同的含义

3. 请求信息查看



在请求发起信息中的一些标签含义：

- 标头：请求header信息
- 文本：请求参数的文本形式
- 十六进制：请求参数的十六进制形式
- 表单：请求参数的表单形式
- 原始数据：请求发起的所有信息，在这里可以看到请求发起的所有信息

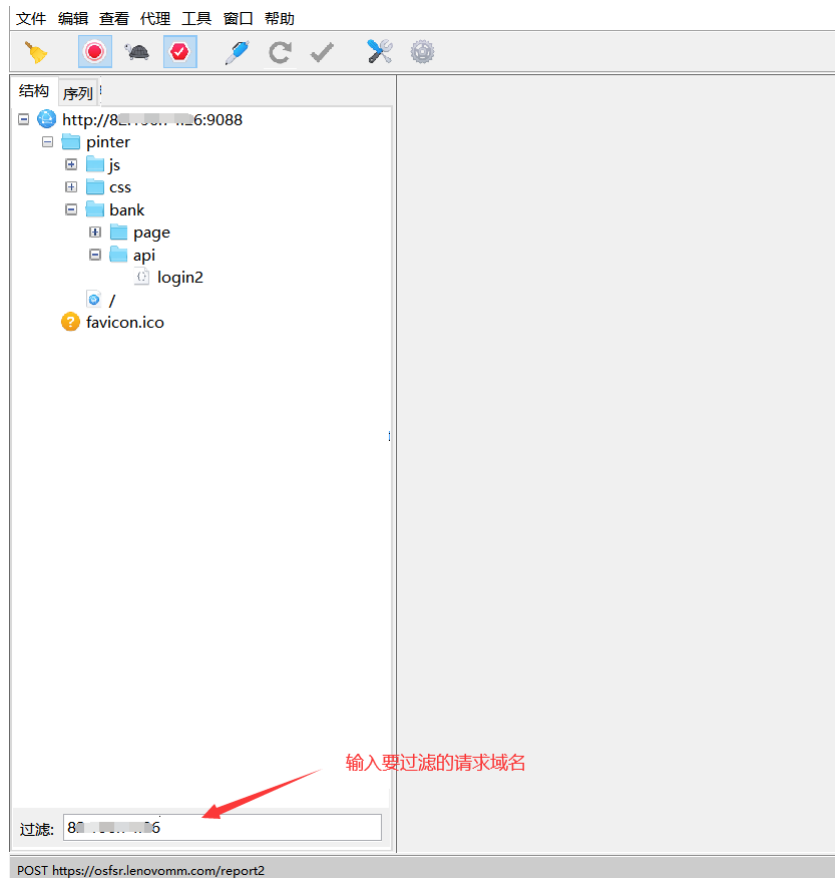
在请求响应信息中的一些标签含义

- 标头：服务器响应的header信息
- 文本：响应内容的文本形式
- 十六进制：响应内容的十六进制形式
- JavaScript：响应内容的js形式
- JSON：响应内容的JSON格式形式
- JSON文本：响应内容的JSON格式文本形式
- 原始数据：服务器响应信息的所有内容都在这里

7.请求过滤

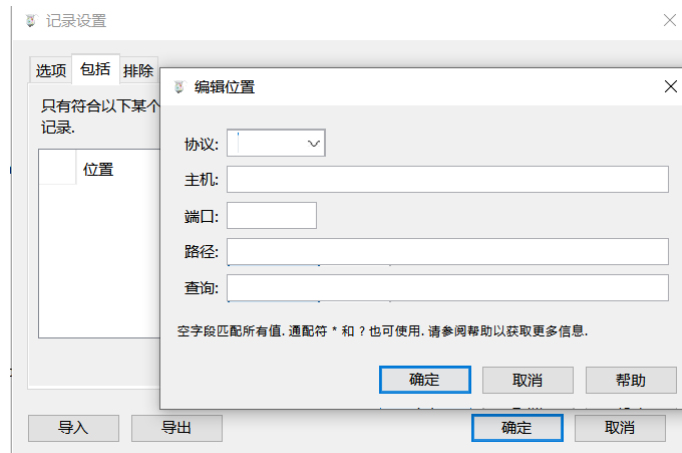
通常情况下我们可能会抓包非常多的请求，那么为了方便我们查找自己的目标请求，可以针对性的设置一些过滤条件

- 快捷过滤方式



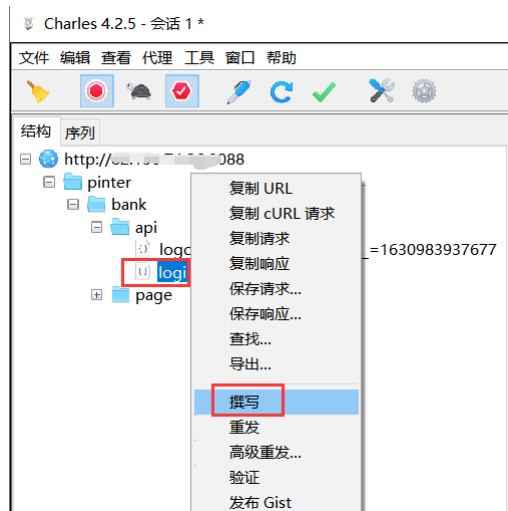
- 多条件过滤

上方工具栏的代理—记录设置—包括—添加

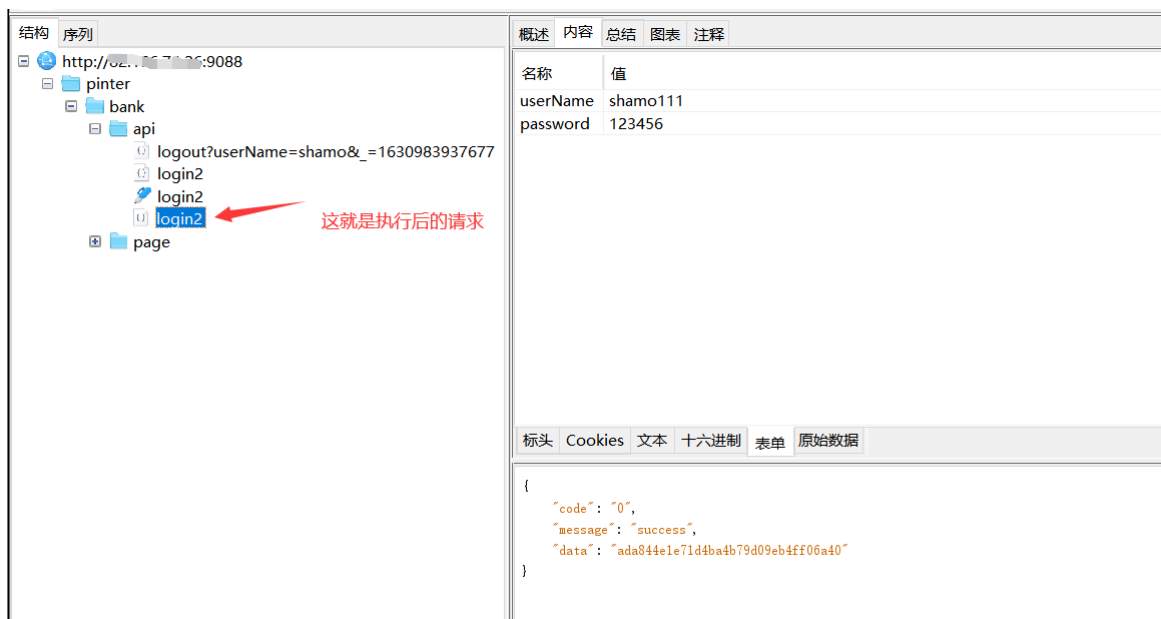
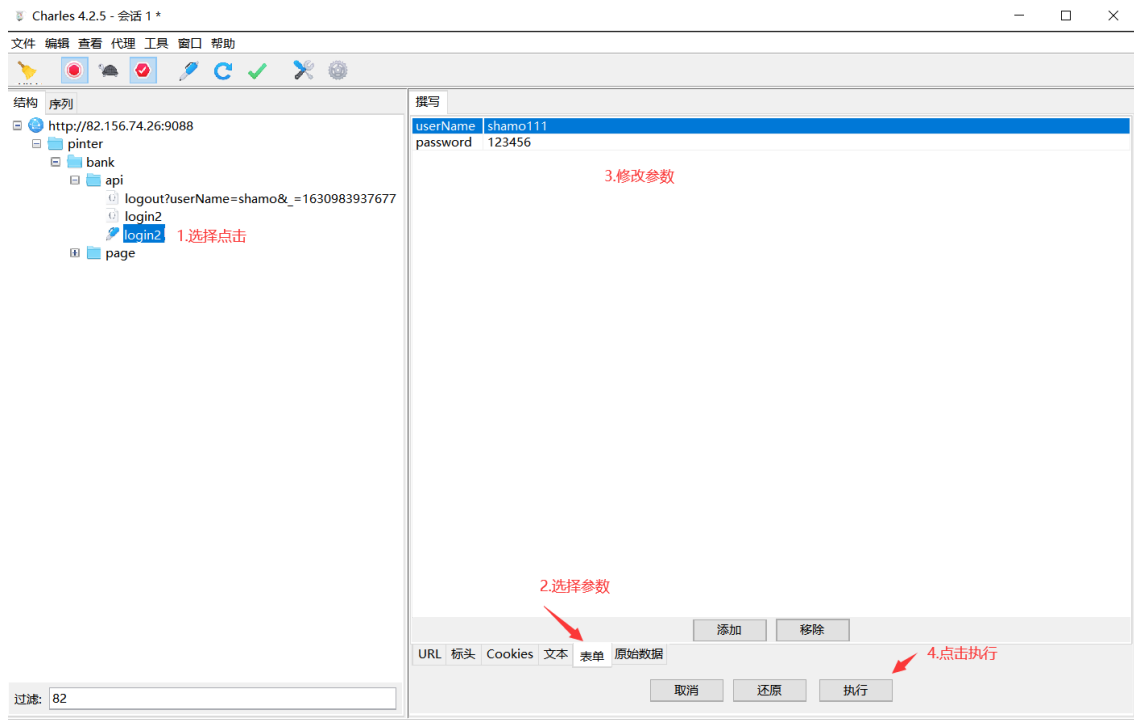


8.接口请求调试

1. 选中要调试的请求，右键-->撰写



2. 修改参数并执行



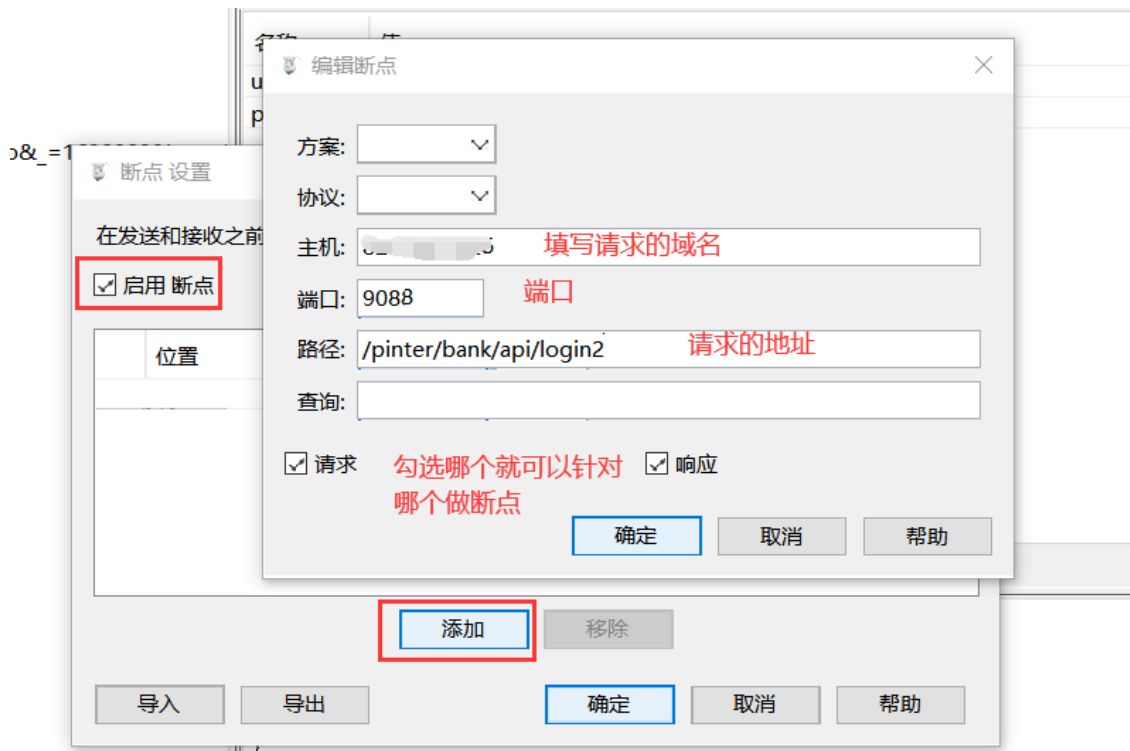
这种方式也可以理解为篡改请求报文，再次发起请求的方式

9.断点

断点的意思就是请求发起或者响应时会被暂停，使用者可以根据自己的需求编辑篡改请求报文或者响应报文，以达到不可告人的目的

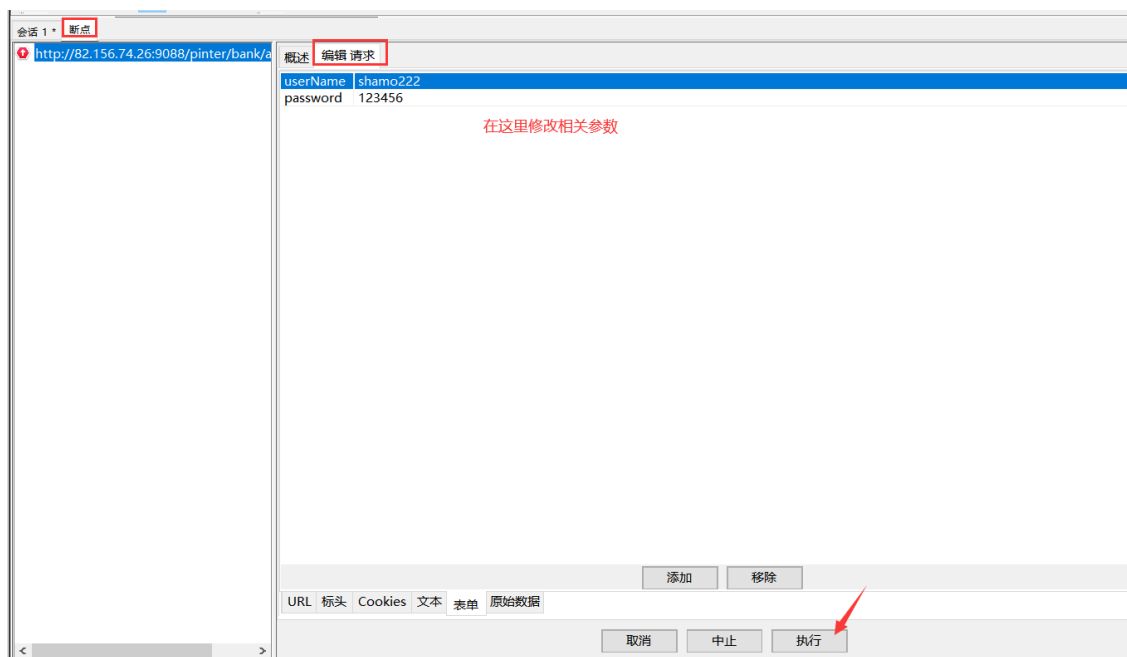
- 断点设置

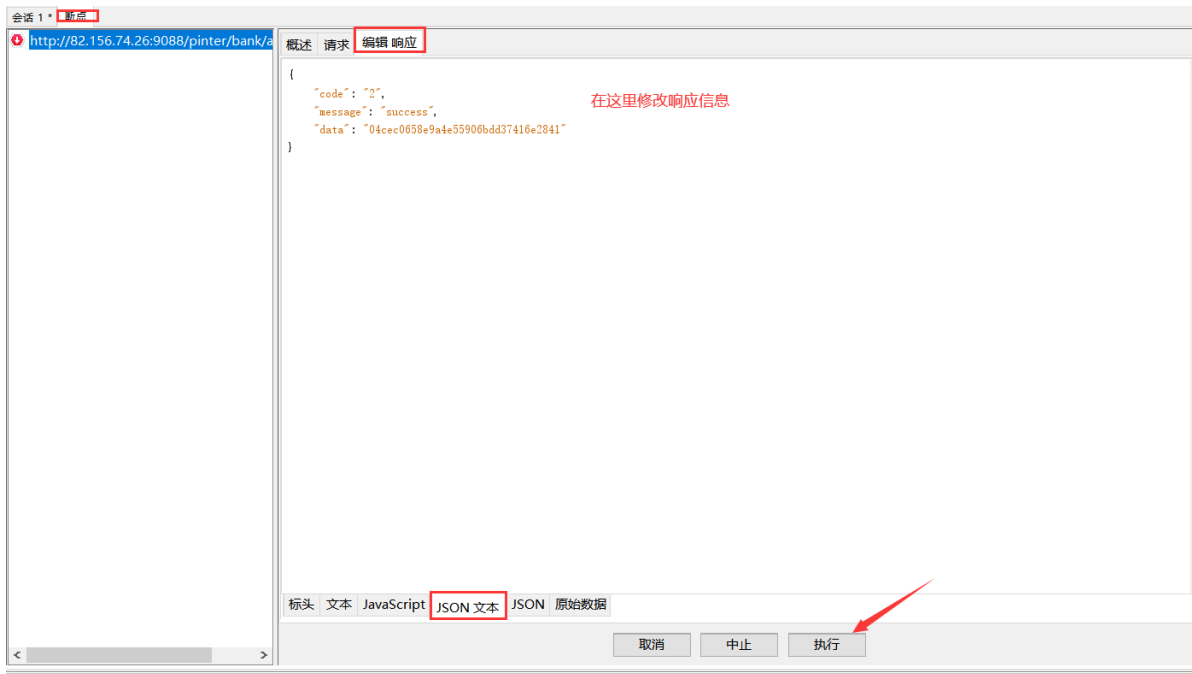
在上方菜单栏代理-->断点设置，点开后添加，添加后当捕获的请求命中设置的各项信息，就会被暂停



- 执行请求，篡改报文

要么在页面上重新访问，要么再charles里右键请求点击重发，都可以进入编辑报文的界面





10.接口请求映射

在断点中篡改报文并不是很方便，每次都得改，那么为了一劳永逸的修改某个接口的响应信息，可以用过接口映射的方式，

将指定接口的响应信息指向我期望的本地文件或者远程接口

1. 本地映射

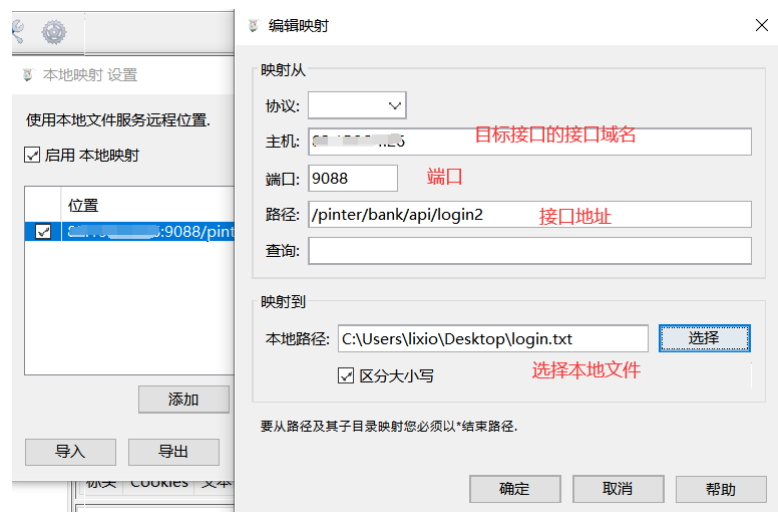
本地映射的意思就是将某个接口的响应内容指向本地文件，因此我们先创建一个本地文件，针对码同学全栈接口项目的

登录接口做本地映射，创建一个txt文件，写入如下内容，这内容里改了响应信息中的code是500：

```
{"code": "500", "message": "success", "data": "04cec0658e9a4e55906bdd37416e2841"}
```

设置本地映射

在上方菜单栏工具-->本地映射



然后重发(回放)目标接口，查看响应

```
HTTP/1.1 200 OK
Content-Length: 76
Content-Type: text/plain
X-Charles-Map-Local: C:\Users\lixio\Desktop\login.txt
Connection: keep-alive
```

```
{ "code": "500", "message": "success", "data": "04cec0658e9a4e55906bdd37416e2841" }
```

2. 远程映射

先来看一个场景

app在线上出现了问题，你想测下在测试环境是否有问题，其实就是你想用线上的包发请求时不要发线上，发到测试环境。

或者

有个app需要测试，测试环境已经测试完成，需要到灰度环境测试。但是，现在没有灰度环境对应的app，那怎么办

有两种方法能解决：

一种是让开发给你重新打包，把里面请求线上的地址换成测试的地址

一种是在charles里做域名映射

如果开发忙没有时间给你打包，那我们就可以直接在charles里域名映射

设置远程映射

在上方菜单栏工具-->远程映射



访问原域名都会去访问目标域名了

11. 重写

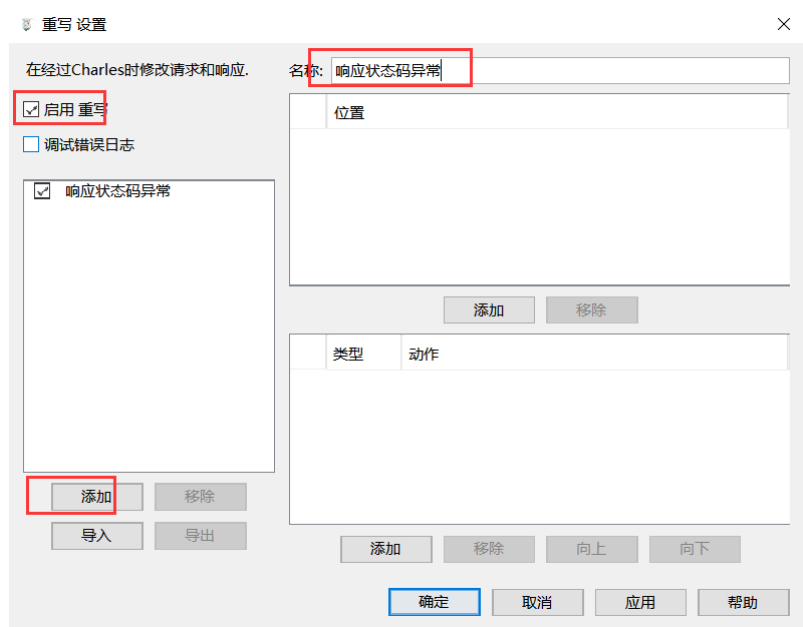
重写功能适合对某一类网络请求进行一些正则替换，以达到修改结果的目的。

可以重写接口所有元素的内容：`header`、`host`、`url`、`path`、`query param`、`response status`、`body`。

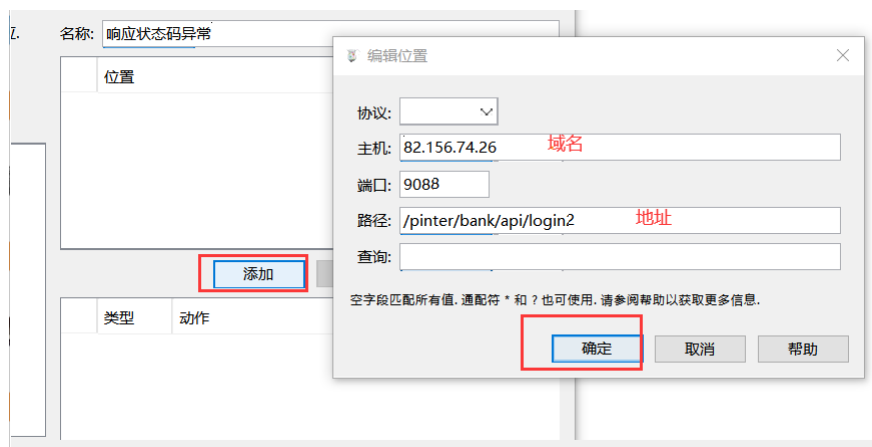
比如我要模拟响应状态码的异常情况

在上方菜单栏工具-->重写

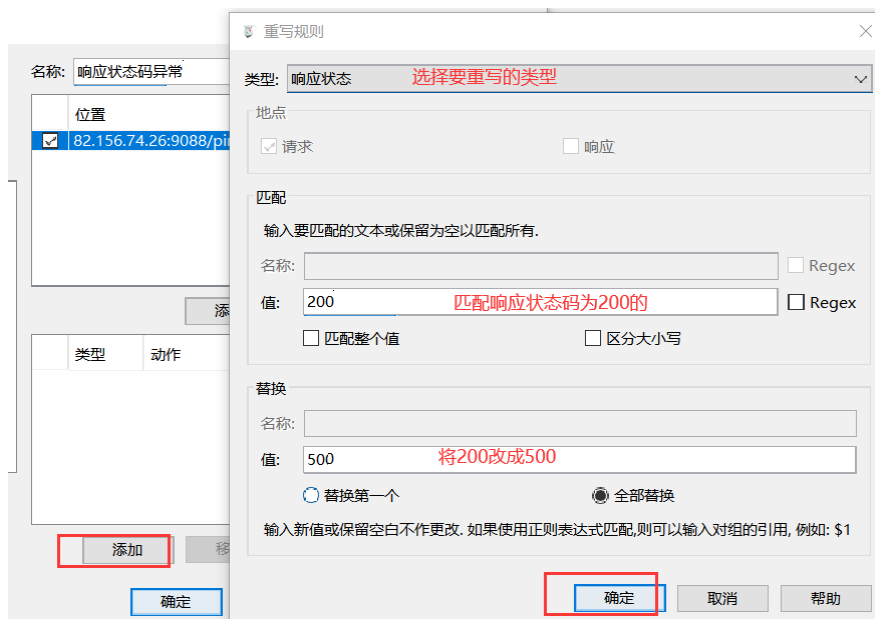
1. 添加一个规则



2. 添加匹配请求



3. 添加重写规则



4. 确定以后，在页面访问，登录接口响应状态码会被改成500



5. 除了针对响应状态码，其他接口的信息也可以被重写，可以在重写规则那里进行选择

