

# iOS开发工具-网络封包分析工具Charles

Dec 11th, 2013



## 简介

本文为InfoQ中文站特供稿件，首发地址为：[文章链接](#)。如需转载，请与InfoQ中文站联系。

[Charles](#)是在Mac下常用的截取网络封包的工具，在做iOS开发时，我们为了调试与服务器端的网络通讯协议，常常需要截取网络封包来分析。Charles通过将自己设置成系统的网络访问代理服务器，使得所有的网络访问请求都通过它来完成，从而实现了网络封包的截取和分析。

Charles是收费软件，可以免费试用30天。试用期过后，未付费的用户仍然可以继续使用，但是每次使用时间不能超过30分钟，并且启动时将会有10秒种的延时。

因此，该付费方案对广大用户还是相当友好的，即使你长期不付费，也能使用完整的软件功能。只是当你需要长时间进行封包调试时，会因为Charles强制关闭而遇到影响。

Charles主要的功能包括：

1. 支持SSL代理。可以截取分析[SSL](#)的请求。
2. 支持流量控制。可以模拟慢速网络以及等待时间（latency）较长的请求。
3. 支持AJAX调试。可以自动将json或xml数据格式化，方便查看。
4. 支持AMF调试。可以将Flash Remoting 或 Flex Remoting信息格式化，方便查看。
5. 支持重发网络请求，方便后端调试。
6. 支持修改网络请求参数。
7. 支持网络请求的截获并动态修改。
8. 检查HTML，CSS和RSS内容是否符合[W3C标准](#)。

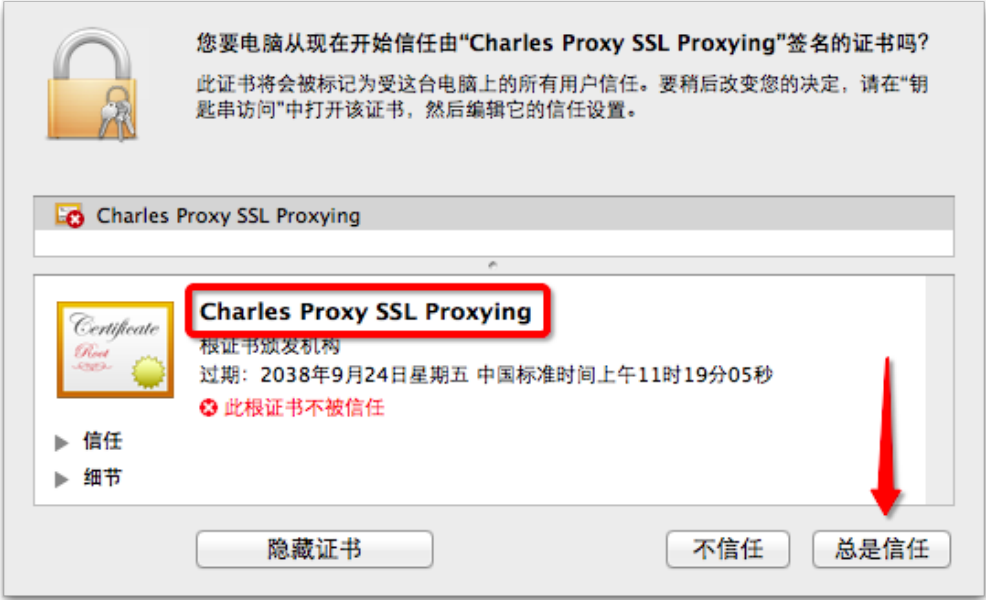
## 安装Charles

去Charles的官方网站（<http://www.charlesproxy.com>）下载最新版的Charles安装包，是一个dmg后缀的文件。打开后将Charles拖到Application目录下即完成安装。

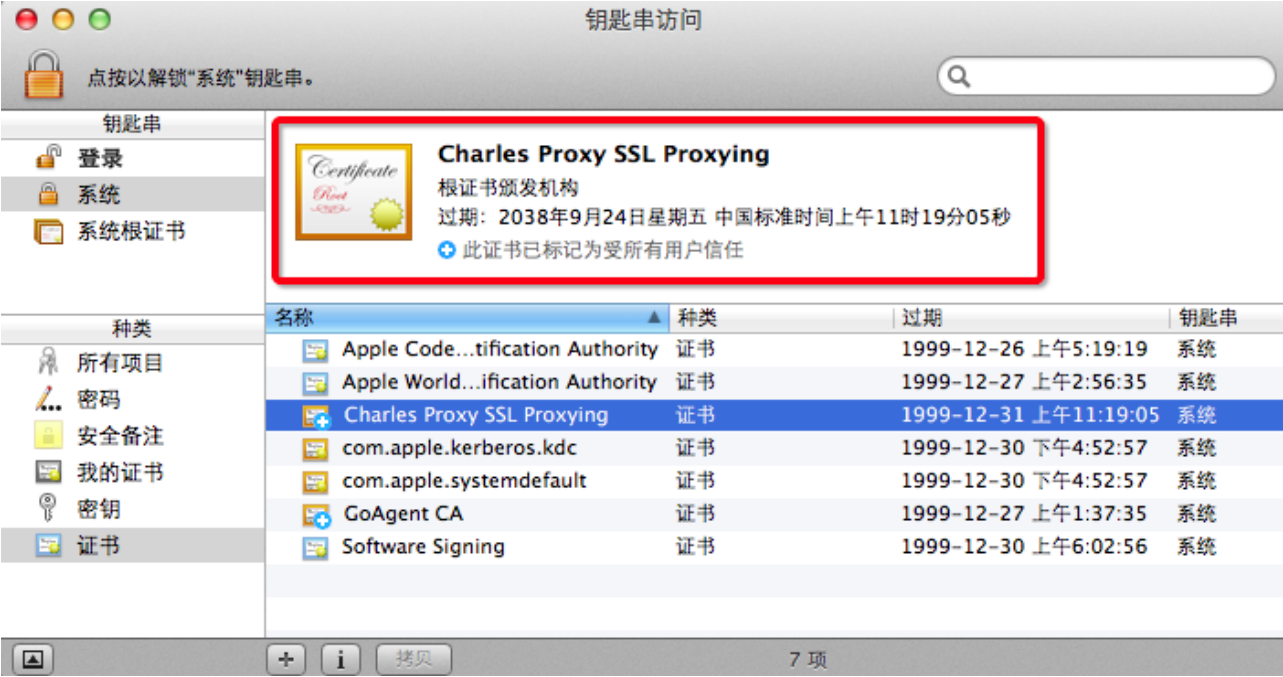
## 安装SSL证书

如果你需要截取分析SSL协议相关的内容。那么需要安装Charles的CA证书。具体步骤如下：

- 1. 去 <http://www.charlesproxy.com/ssl.zip> 下载CA证书文件。
- 2. 解压该zip文件后，双击其中的.crt文件，这时候在弹出的菜单中选择“总是信任”，如下所示：



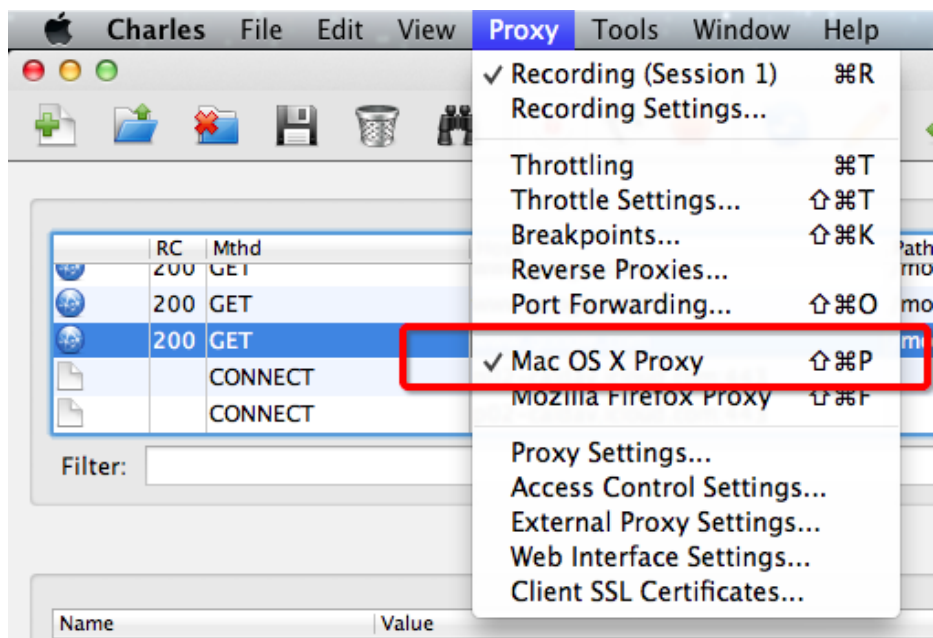
- 3. 从钥匙串访问中即可看到添加成功的证书。如下所示：



## 将Charles设置成系统代理

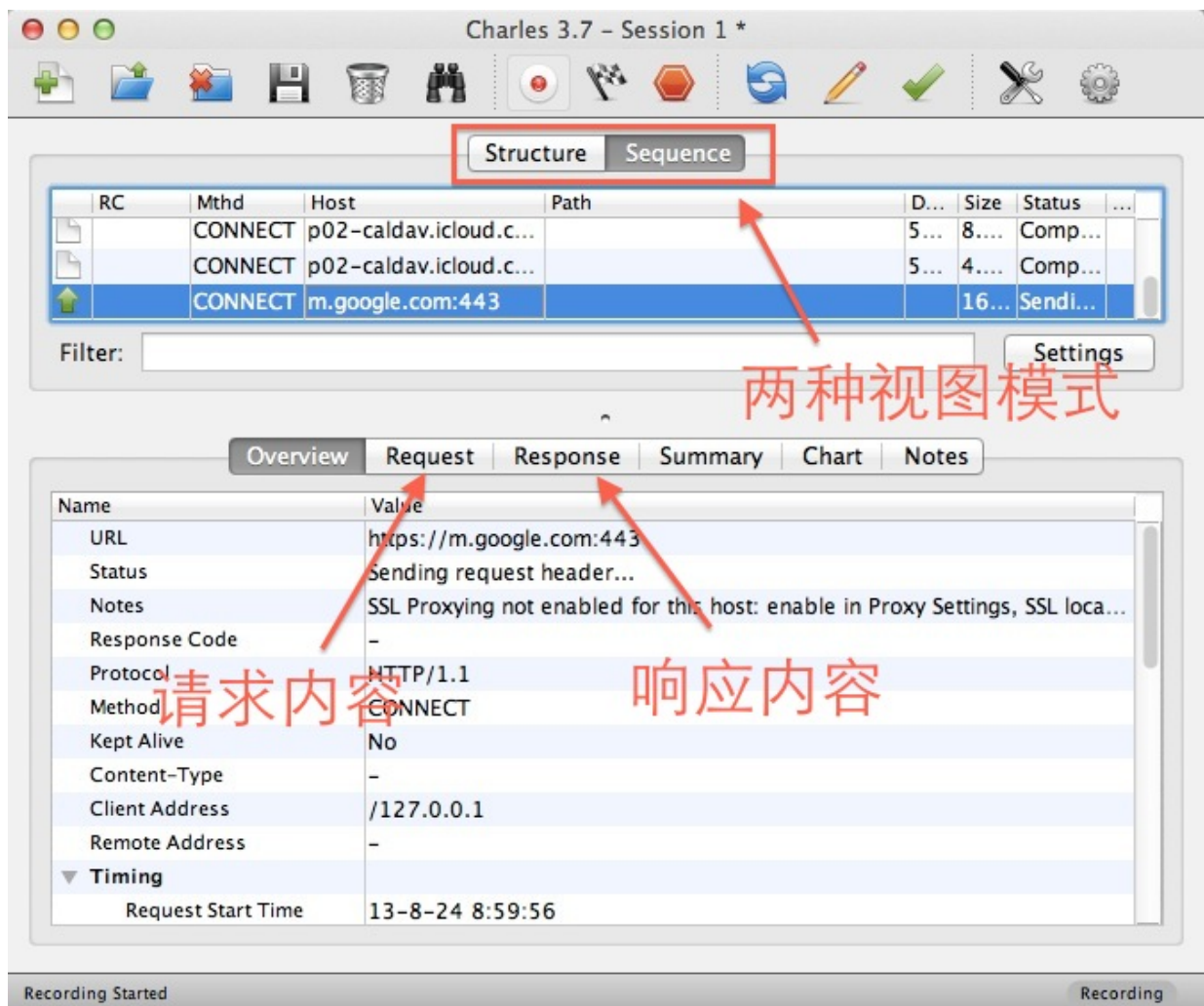
之前提到，Charles是通过将自己设置成代理服务器来完成封包截取的，所以使用Charles的第一步是将其设置成系统的代理服务器。

启动Charles后，第一次Charles会请求你给它设置系统代理的权限。你可以输入登录密码授予Charles该权限。你也可以忽略该请求，然后在需要将Charles设置成系统代理时，选择菜单中的“Proxy”->“Mac OS X Proxy”来将Charles设置成系统代理。如下所示：



之后，你就可以看到源源不断的网络请求出现在Charles的界面中。

## Charles主界面介绍



Charles主要提供2种查看封包的视图，分别名为“Structure”和“Sequence”。

1. Structure视图将网络请求按访问的域名分类。
2. Sequence视图将网络请求按访问的时间排序。

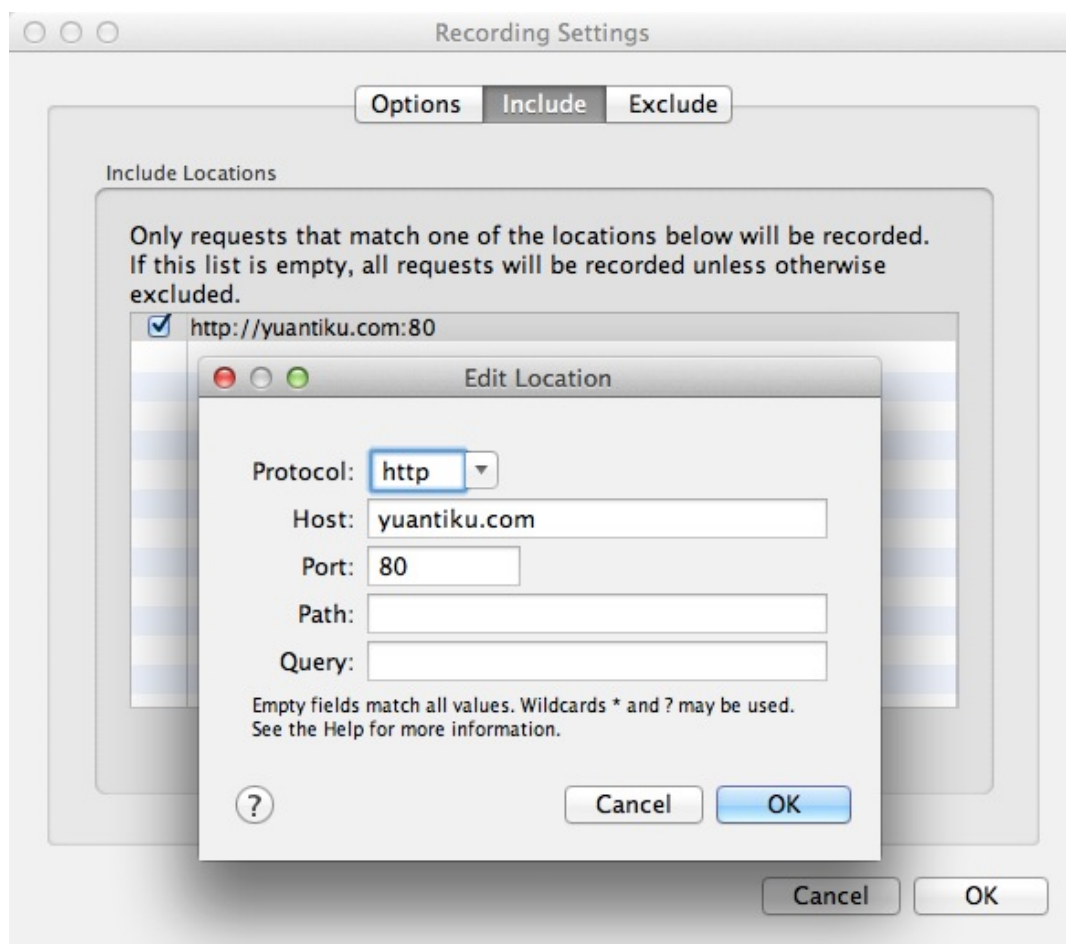
大家可以根据具体的需要在这两种视图之前来回切换。

对于某一个具体的网络请求，你可以查看其详细的请求内容和响应内容。如果响应内容是JSON格式的，那么Charles可以自动帮你将JSON内容格式化，方便你查看。

## 过滤网络请求

通常情况下，我们需要对网络请求进行过滤，只监控向指定目录服务器上发送的请求。对于这种需求，我们有2种办法。

1. 在主界面的中部的Filter栏中填入需要过滤出来的关键字。例如我们的服务器的地址是：<http://yuantiku.com> ,那么只需要在Filter栏中填入yuantiku即可。
2. 在Charles的菜单栏选择”Proxy”->”Recording Settings”，然后选择Include栏，选择添加一个项目，然后填入需要监控的协议，主机地址，端口号。这样就可以只截取目标网站的封包了。如下图所示：



通常情况下，我们使用方法1做一些临时性的封包过滤，使用方法2做一些经常性的封包过滤。

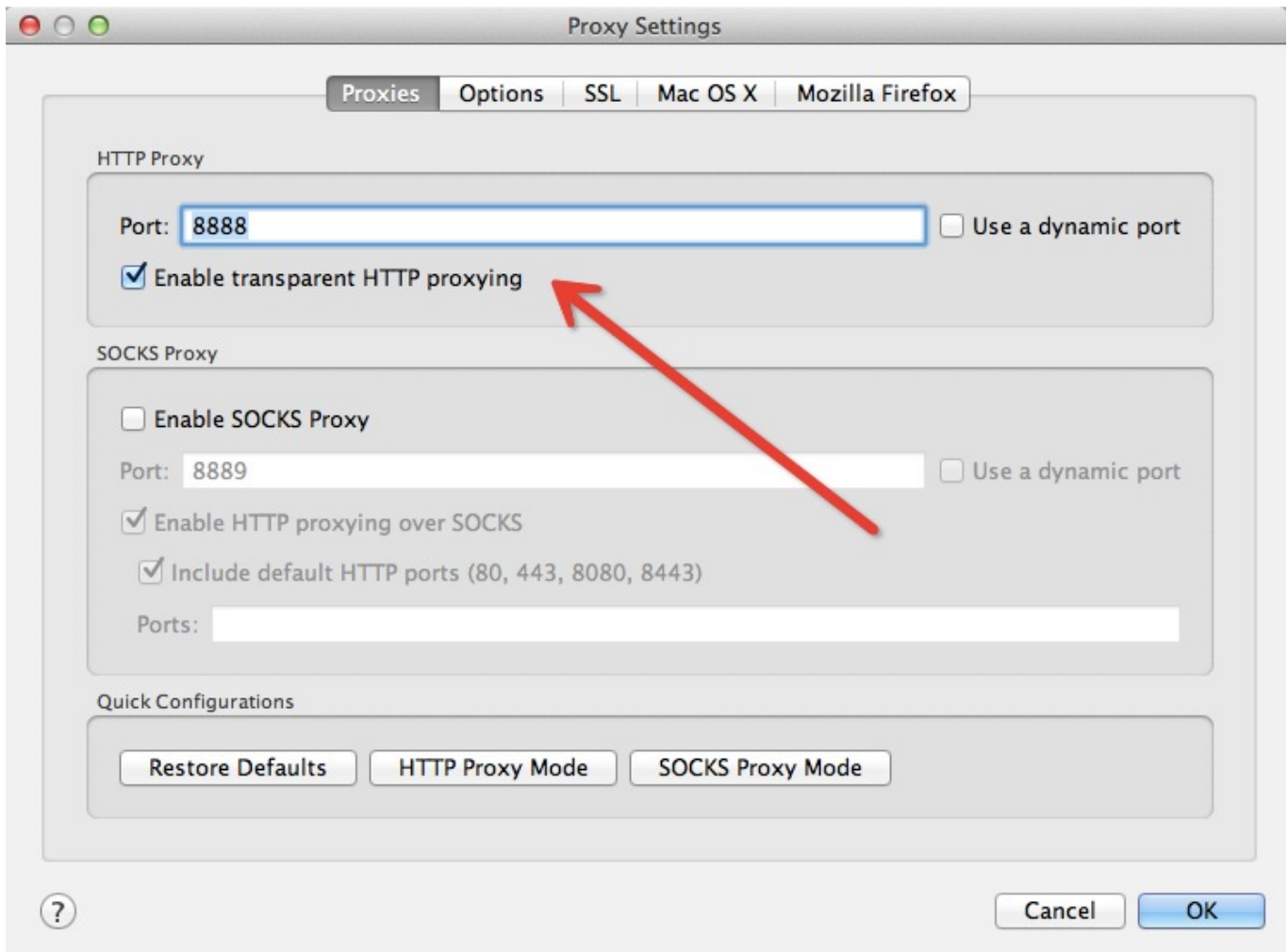
## 截取iPhone上的网络封包

Charles通常用来截取本地上的网络封包，但是当我们需要时，我们也可以用来截取其它设备上的网络请求。下面我就以iPhone为例，讲解如何进行相应操作。

### Charles上的设置

要截取iPhone上的网络请求，我们首先需要将Charles的代理功能打开。在Charles的菜单栏上选择“Proxy”->“Proxy Settings”，填入代理端口8888，并且勾选”Enable transparent HTTP proxying” 就完成了在Charles上的设置。如下图所示：





## iPhone上的设置

首先我们需要获取Charles运行所在电脑的IP地址，打开Terminal，输入`ifconfig en0`，即可获得该电脑的IP，如下图所示：

```
➔ ~ ifconfig en0
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether 84:38:35:5c:2a:88
    inet6 fe80::8638:35ff:fe5c:2a88%en0 prefixlen 64 scopeid 0x4
    inet 192.168.1.105 netmask 0xffffffff broadcast 192.168.1.255
    media: autoselect
    status: active
```

在iPhone的“设置”->“无线局域网”中，可以看到当前连接的wifi名，通过点击右边的详情键，可以看到当前连接上的wifi的详细信息，包括IP地址，子网掩码等信息。在其最底部有“HTTP代理”一项，我们将其切换成手动，然后填上Charles运行所在的电脑的IP，以及端口号8888，如下图所示：

⬅ 无线局域网 TP-LINK\_PocketAP\_603

技术域

客户端 ID

续租

HTTP 代理

关闭

手动

自动

服务器

192.168.1.105

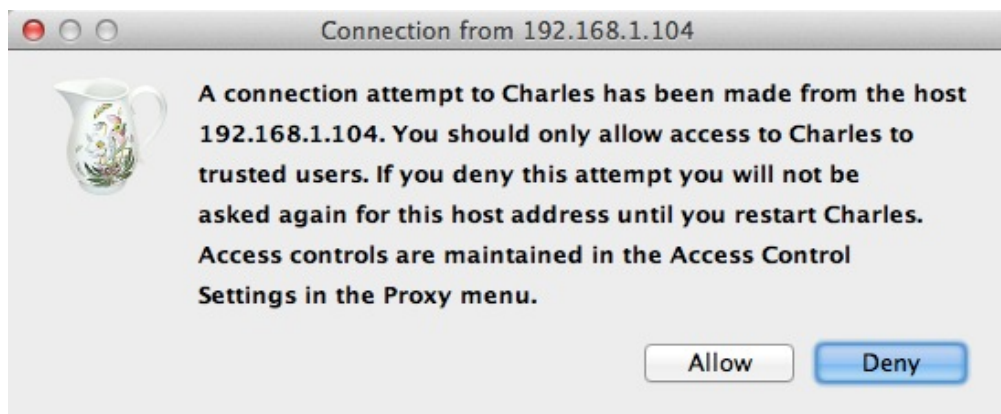
端口

8888

鉴定

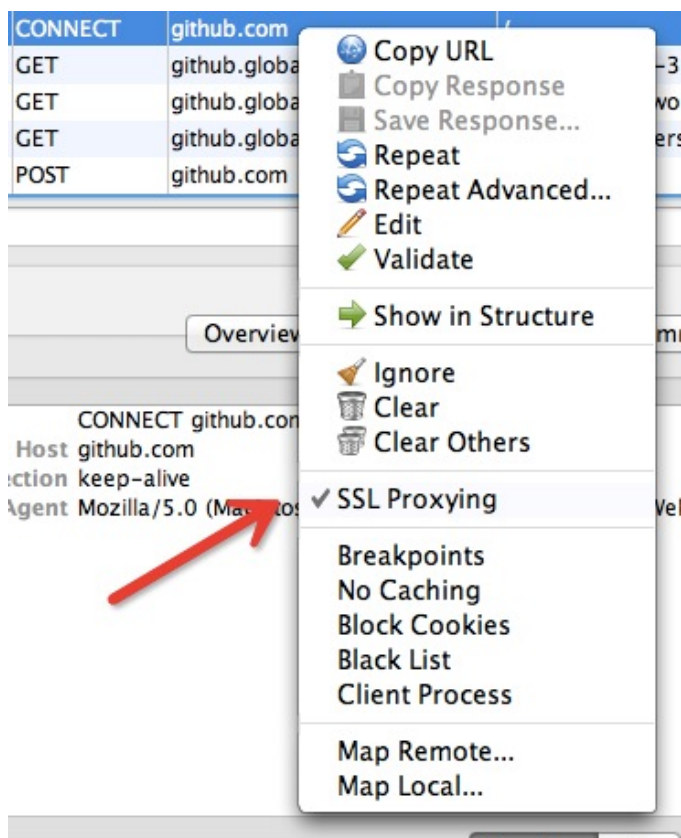


设置好之后，我们打开iPhone上的任意需要网络通讯的程序，就可以看到Charles弹出iPhone请求连接的确认菜单（如下图所示），点击“Allow”即可完成设置。



## 截取SSL信息

Charles默认并不截取SSL的信息，如果你想对截取某个网站上的所有SSL网络请求，可以在该请求上右击，选择SSL proxy，如下图所示：

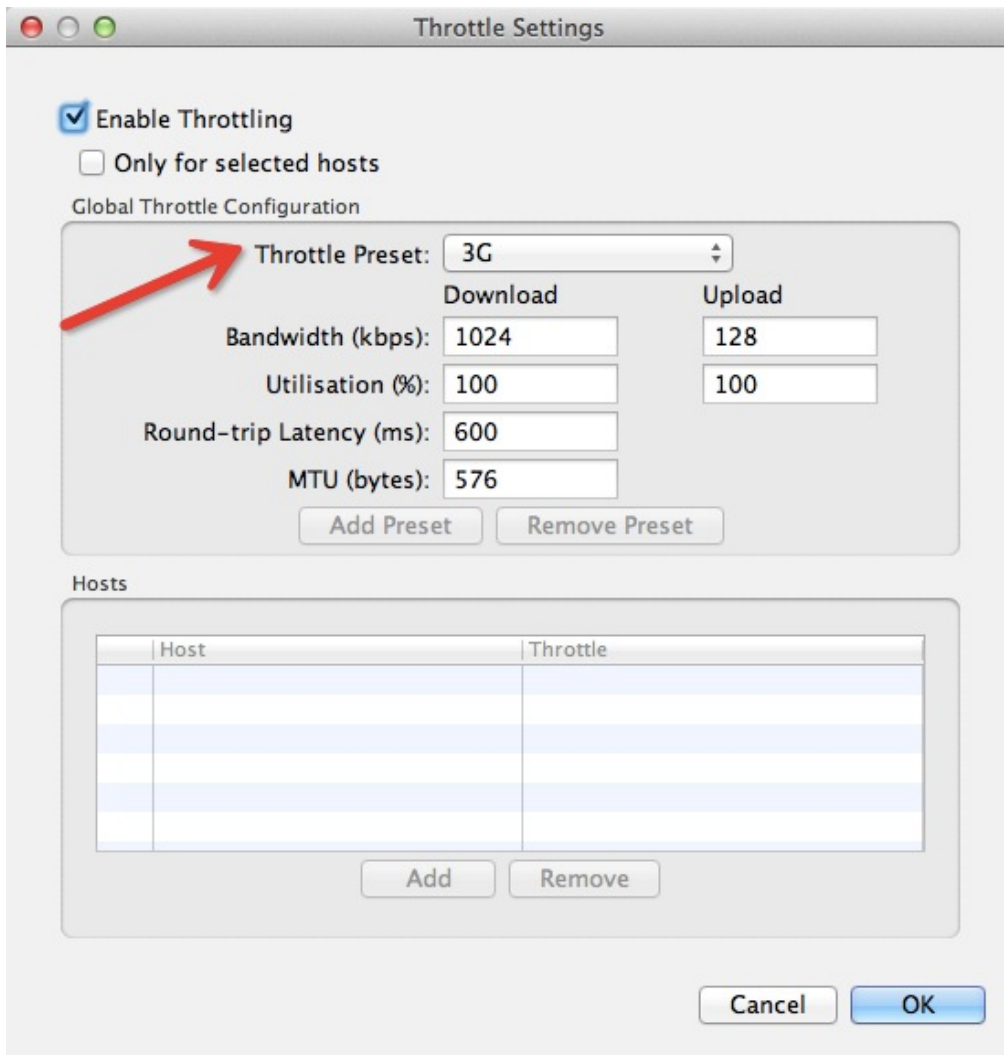


这样，对于该Host的所有SSL请求可以被截取到了。

## 模拟慢速网络

在做iPhone开发的时候，我们常常需要模拟慢速网络或者高延迟的网络，以测试在移动网络下，应用的表现是否正常。Charles对此需求提供了很好的支持。

在Charles的菜单上，选择“Proxy”->“Throttle Setting”项，在之后弹出的对话框中，我们可以勾选上“Enable Throttling”，并且可以设置Throttle Preset的类型。如下图所示：

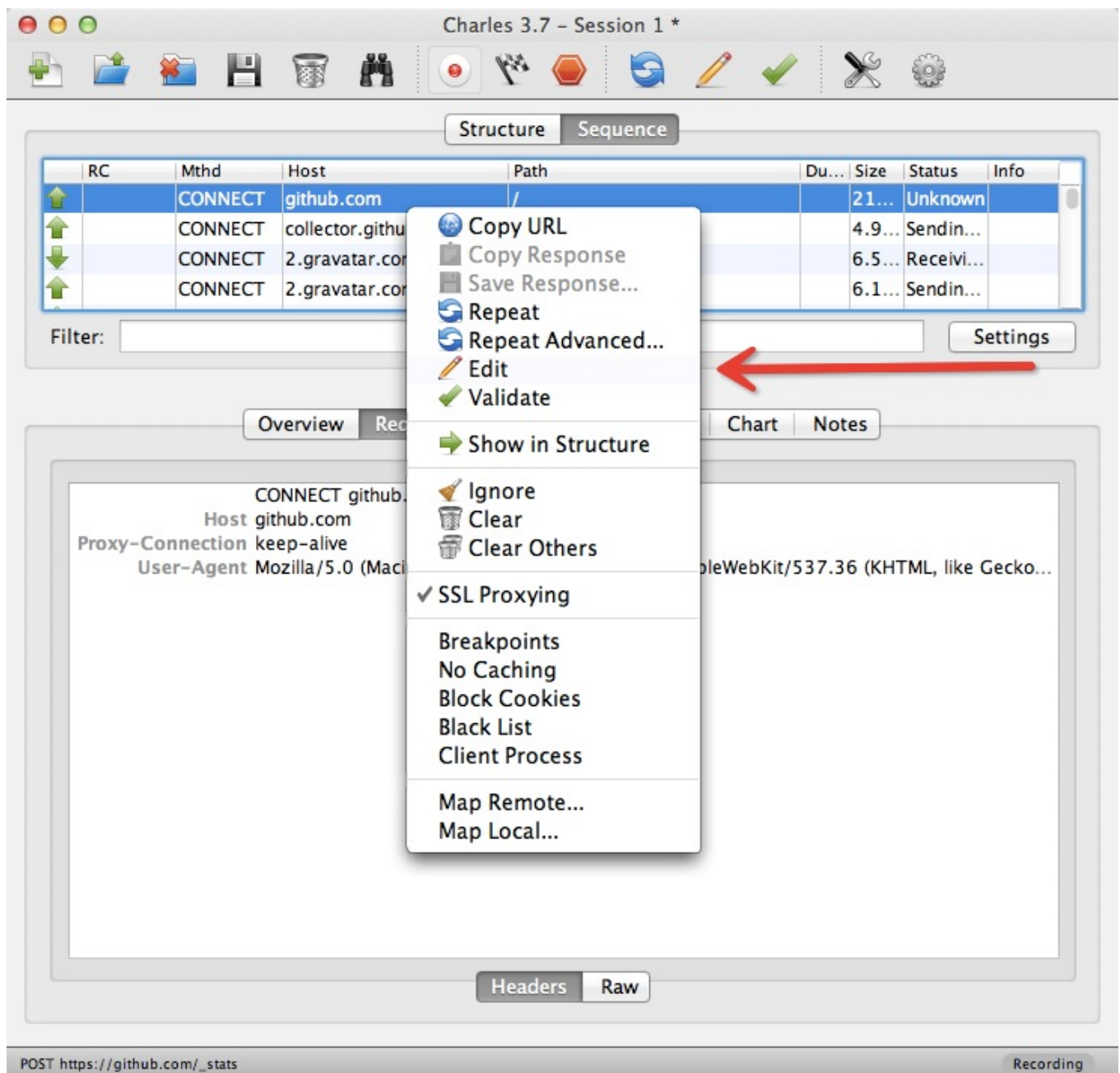


如果我们只想模拟指定网站的慢速网络，可以再勾选上图中的”Only for selected hosts”项，然后在对话框的下半部分设置中增加指定的hosts项即可。

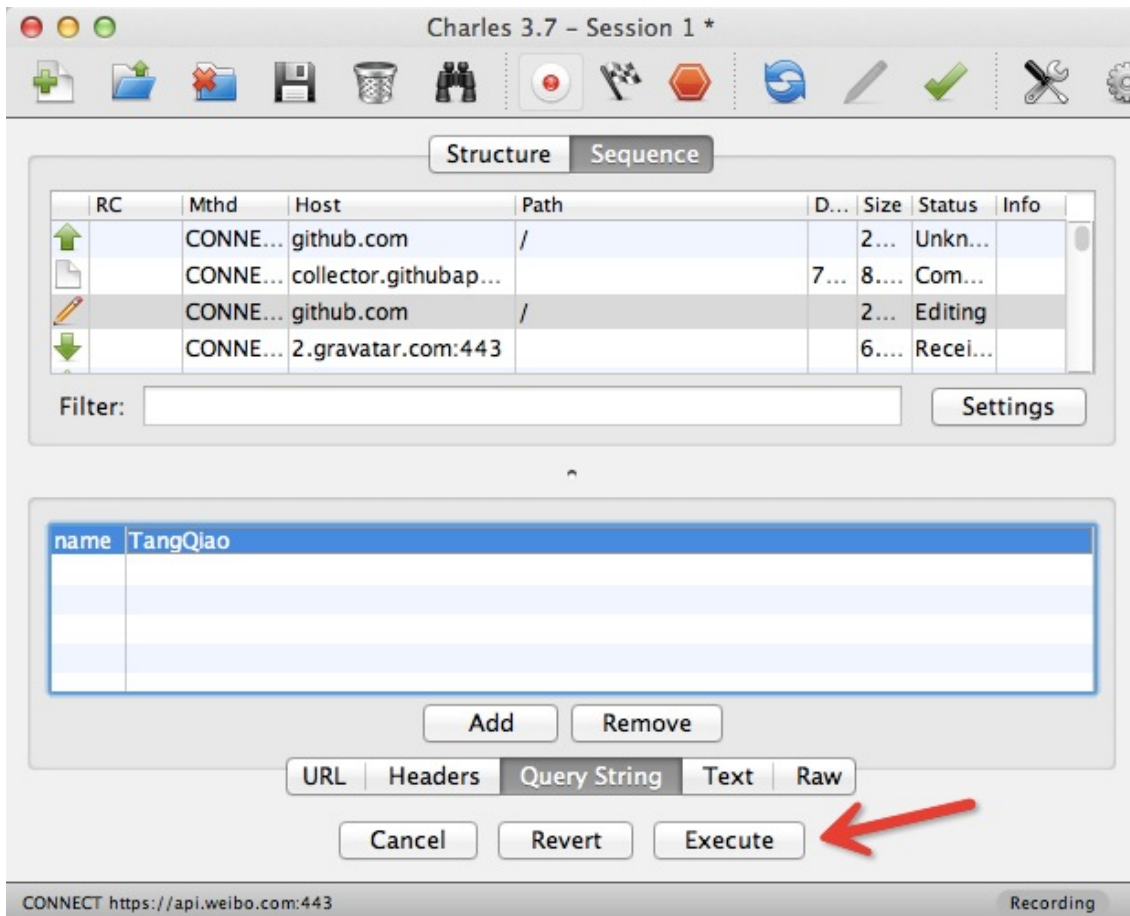
## 修改网络请求内容

有些时候为了调试服务器的接口，我们需要反复尝试不同参数的网络请求。Charles可以方便地提供网络请求的修改和重发功能。只需要在以往的网络请求上点击右键，选择“Edit”，即可创建一个可编辑的网络请求。如下所示：





我们可以修改该请求的任何信息，包括url地址，端口，参数等，之后点击“Execute”即可发送该修改后的网络请求（如下图所示）。Charles支持我们多次修改和发送该请求，这对于我们和服务端调试接口非常方便。



## 总结

通过Charles软件，我们可以很方便地在日常开发中，截取和调试网络请求内容，分析封包协议以及模拟慢速网络。用好Charles可以极大的方便我们对于带有网络请求的App的开发和调试。

参考链接：

1. [Charles主要的功能列表](#)
2. [Charles官网](#)

© 2015 - 唐巧 - 基于 [Octopress](#) , 感谢 [GitCafe](#) 为本站提供存储空间

# 总结

第一步:下载,安装

第二步:wifi和手机在同一个局域网下

第三步:charles在osx下设置代理,(端口8888,勾选http代理)

第四步:android手机,wifi>wifi名称,长按>修改网络,

>显示高级选项>代理设置为手动,填入电脑的>ip和>port,保存

第五步:charles开启抓包

第六步:手机端启动应用,观察charles包记录

charles

- 可以对请求的头,体,参数查看.
- 根据规则过滤监听
- 查看响应的格式(自带json查看器)
- 保存响应的文件(保存html,jpg,json等文件)

2015-01-18\_01\_28\_14

