* Charles 的简介
* 如何安装 Charles
* 将 Charles 设置成系统代理
* Charles 主界面介绍
* 过滤网络请求
* 截取 iPhone 上的网络封包
* 截取 Https 通讯信息
* 模拟慢速网络
* 修改网络请求内容
* 给服务器做压力测试
* 修改服务器返回内容
* 总结

**简介**

[Charles](http://www.charlesproxy.com/) 是网络封包截取工具。

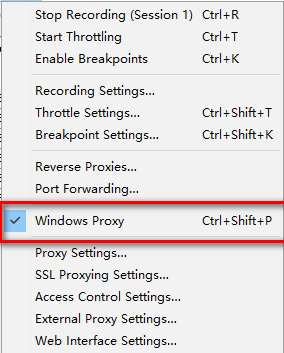
**Charles 通过将自己设置成系统的网络访问代理服务器，使得所有的网络访问请求都通过它来完成，从而实现了网络封包的截取和分析**。

Charles 也可以用于分析第三方应用的通讯协议。配合 Charles 的 SSL 功能，Charles 还可以分析 Https 协议。

Charles 主要的功能包括：

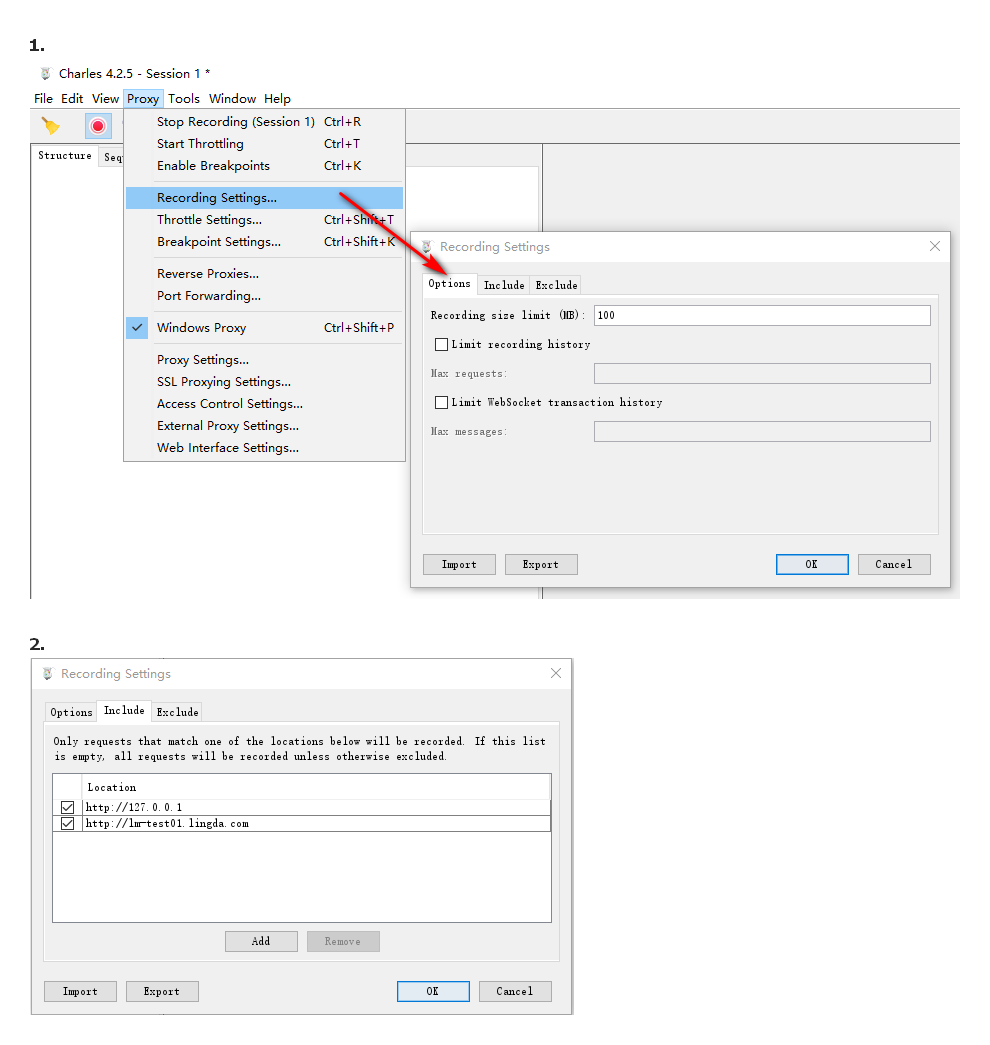
1. 截取 Http 和 Https 网络封包。
2. 支持重发网络请求，方便后端调试。
3. 支持修改网络请求参数。
4. 支持网络请求的截获并动态修改。
5. 支持模拟慢速网络
6. **将 Charles 设置成系统代理**

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，如下图

、

然后，可以看到网络请求出现在 Charles 的界面中。

1. **截取指定网站封包**



Chrome 和 Firefox 浏览器默认**并不使用系统的代理服务器设置**，这导致他们默认不使用 charler做代理，在 Chrome 中设置成使用系统的代理服务器设置即可，或者直接将代理服务器设置成 127.0.0.1:8888 也可达到相同效果。

**Charles 主界面介绍**

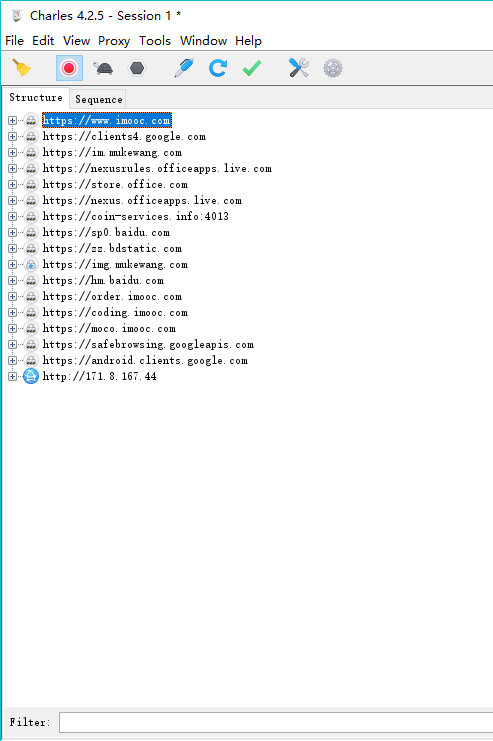
Charles 主要提供两种查看封包的视图，分别名为 “Structure” 和 “Sequence”。

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。

Charles 提供了一个简单的 Filter 功能，可以输入关键字来快速筛选出 URL 中带指定关键字的网络请求。

**过滤网络请求**

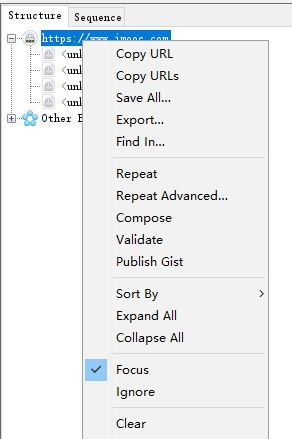
方法一：在主界面的中部的 Filter 栏中填入需要过滤出来的关键字。例如我们的服务器的地址是：http://yuantiku.com , 那么只需要在 Filter 栏中填入



关键字即可。

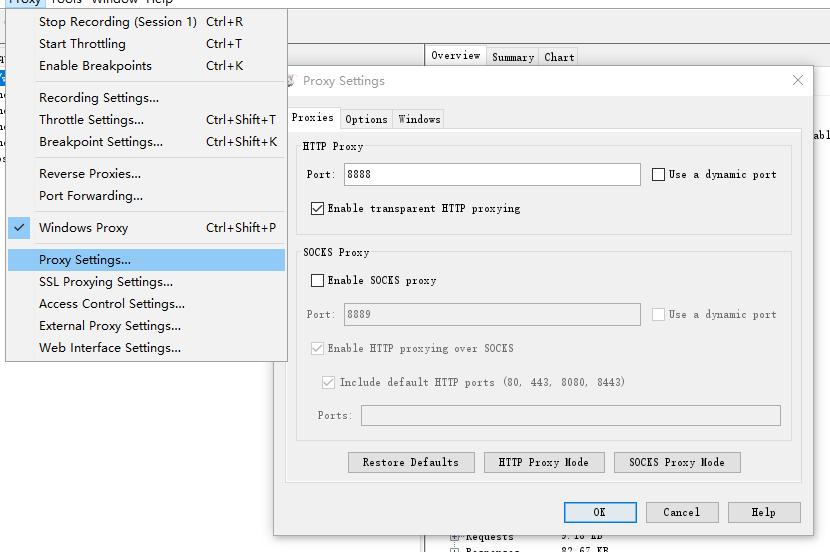
方法二：在 Charles 的菜单栏选择 “Proxy”–>“Recording Settings”，然后选择 Include 栏，选择添加一个项目，然后填入需要监控的协议，主机地址，端口号。

方法三：在想过滤的网络请求上右击，选择 “Focus”，之后在 Filter 一栏勾选上 Focussed 一项，如下图所示：



**Charles 代理设置**

Charles 的代理客户端代理工具，在 Charles 的菜单栏上选择 “Proxy”–>“Proxy Settings”，填入代理端口 8888，并且勾上 “Enable transparent HTTP proxying” 就完成了在 Charles 上的设置。如下图所示:



**修改服务器返回内容**

有些时候我们想让服务器返回一些指定的内容，方便我们调试一些特殊情况。例如列表页面为空的情况，数据异常的情况，部分耗时的网络请求超时的情况等。如果没有 Charles，要服务器配合构造相应的数据显得会比较麻烦。这个时候，使用 Charles 相关的功能就可以满足我们的需求。

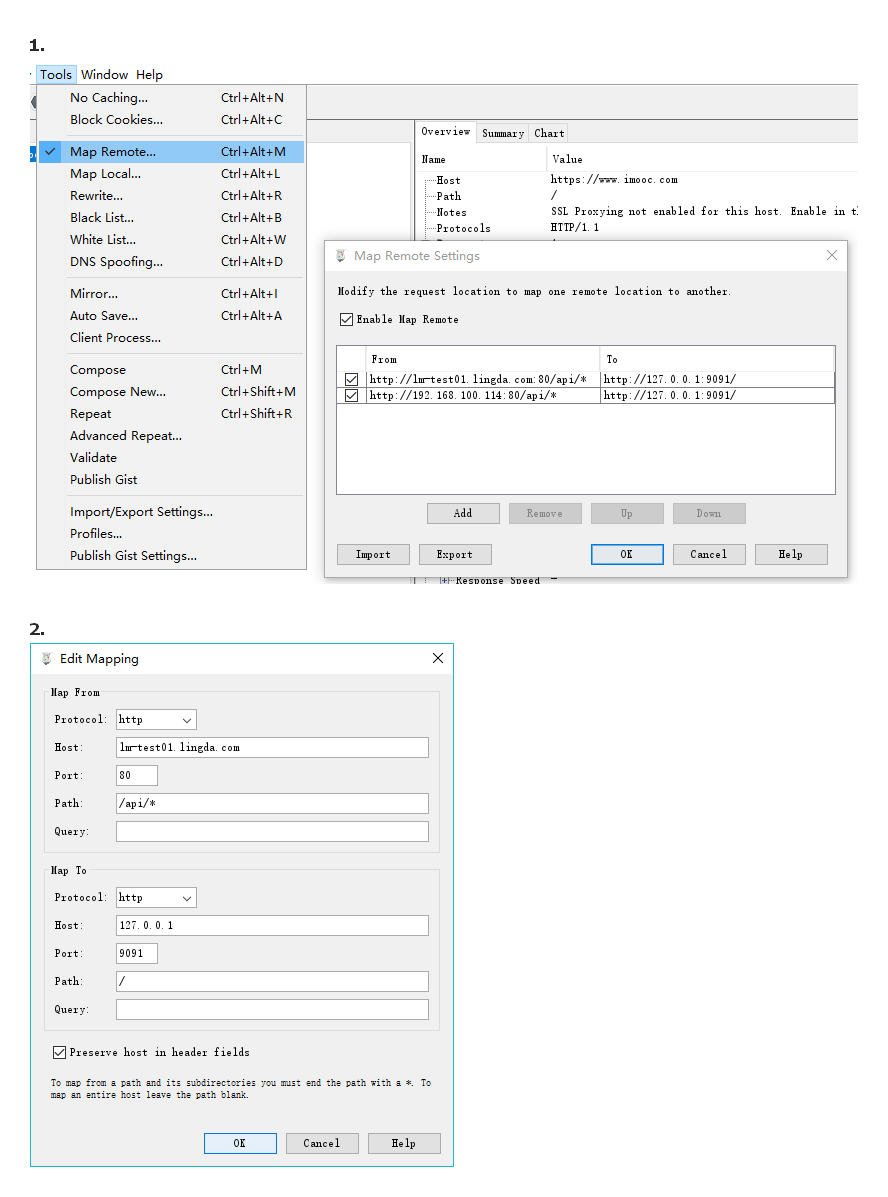
根据具体的需求，Charles 提供了 Map 功能、 Rewrite 功能以及 Breakpoints 功能，都可以达到修改服务器返回内容的目的。这三者在功能上的差异是：

1. Map 功能适合长期地将某一些请求重定向到另一个网络地址或本地文件。
2. Rewrite 功能适合对网络请求进行一些正则替换。
3. Breakpoints 功能适合做一些临时性的修改。

**Map 功能**

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请求地址，Map Local 是将指定的网络请求重定向到本地文件。

在 Charles 的菜单中，选择 “Tools”–>“Map Remote” 或 “Map Local” 即可进入到相应功能的设置页面。



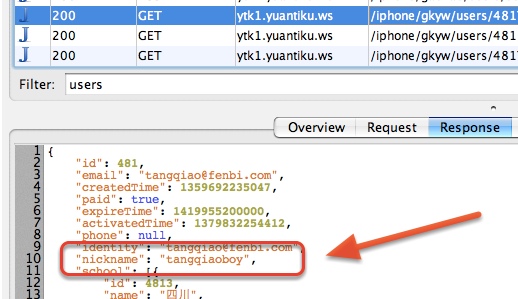
对于 Map Remote 功能，我们需要分别填写网络重定向的源地址和目的地址，对于不需要限制的条件，可以留空。

对于 Map Local 功能，我们需要填写的重定向的源地址和本地的目标文件。对于有一些复杂的网络请求结果，我们可以先使用 Charles 提供的 “Save Response…” 功能，将请求结果保存到本地（如下图所示），然后稍加修改，成为我们的目标映射文件。

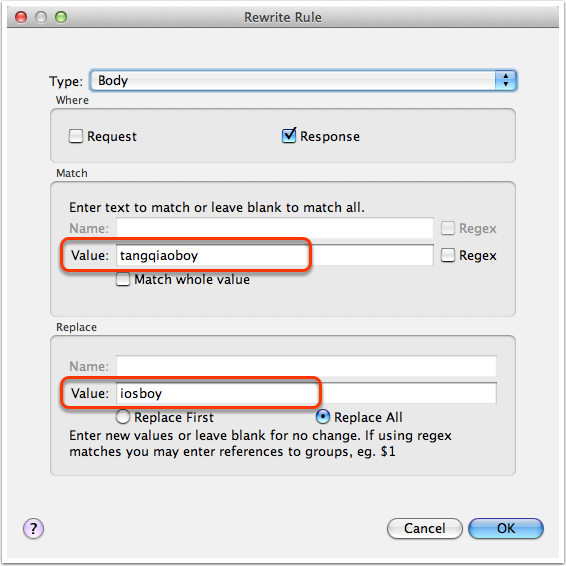
**Rewrite 功能**

Rewrite 功能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。

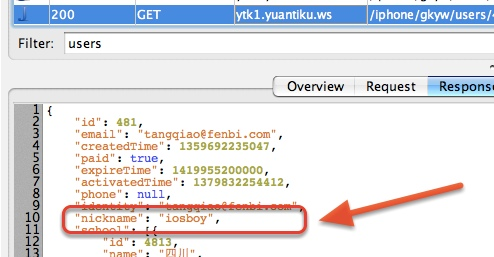
例如，我们的客户端有一个 API 请求是获得用户昵称，而我当前的昵称是 “tangqiaoboy”，如下所示：

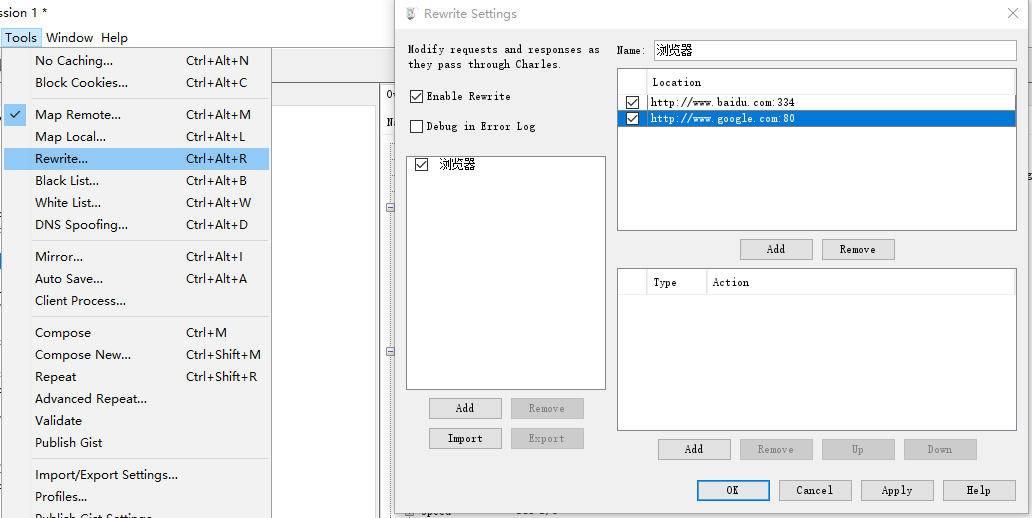


我们想试着直接修改网络返回值，将 tangqiaoboy 换成成 iosboy。于是我们启用 Rewrite 功能，然后设置如下的规则：



完成设置之后，我们就可以从 Charles 中看到，之后的 API 获得的昵称被自动 Rewrite 成了 iosboy，如下图所示：





**Breakpoints 功能**

上面提供的 Rewrite 功能最适合做批量和长期的替换，但是很多时候，我们只是想临时修改一次网络请求结果，这个时候，使用 Rewrite 功能虽然也可以达到目的，但是过于麻烦，对于临时性的修改，我们最好使用 Breakpoints 功能。

当指定的网络请求发生时，Charles 会截获该请求，这个时候，我们可以在 Charles 中临时修改网络请求的返回内容。

下图是我们临时修改获取用户信息的 API，将用户的昵称进行了更改，修改完成后点击 “Execute” 则可以让网络请求继续进行。

需要注意的是，使用 Breakpoints 功能将网络请求截获并修改过程中，整个网络请求的计时并不会暂停，所以长时间的暂停可能导致客户端的请求超时。