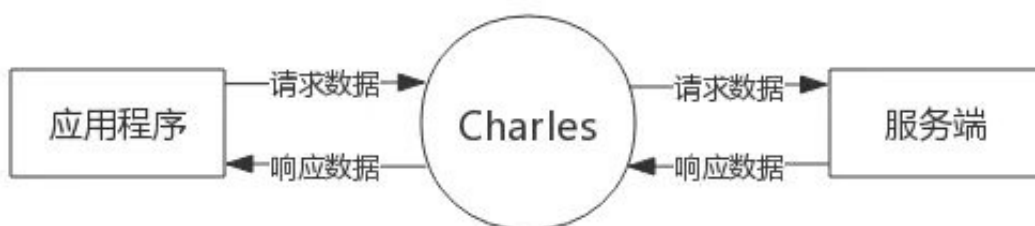


Charles



简介:

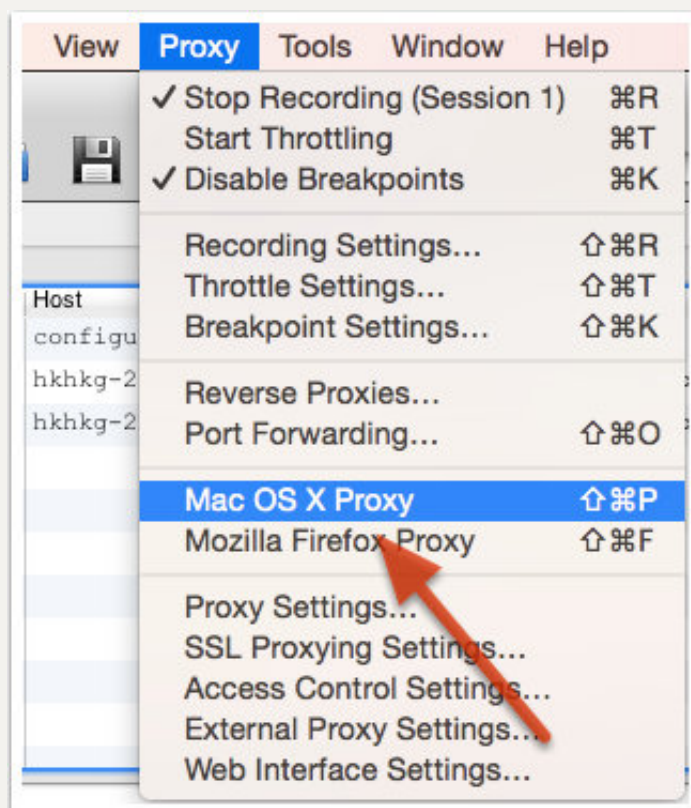
- **Charles** 是在 Mac 下常用的网络封包截取工具，在做移动开发时，我们为了调试与服务器端的网络通讯协议，常常需要截取网络封包来分析
- Charles 通过将自己设置成系统的网络访问代理服务器，使得所有的网络访问请求都通过它来完成，从而实现了网络封包的截取和分析



简单应用：

设置Charles为系统代理：

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：



之后就会抓取电脑的网络请求啦

需要注意的是，Chrome 和 Firefox 浏览器默认并不使用系统的代理服务器设置，而 Charles 是通过将自己设置成代理服务器来完成封包截取的，所以在默认情况下无法截取 Chrome 和 Firefox 浏览器的网络通讯内容。如果你需要截取的话，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:

Structure和Sequence两种预览模式:

- Structure:以域名对请求进行分组，可以很方便地预览同一域名下的请求数据。
- Sequence:以时间顺序显示请求信息，可以最直接的预览请求信息。在此模式下，可以使用Filter过滤请求，针对性分析

抓取手机的网络包:

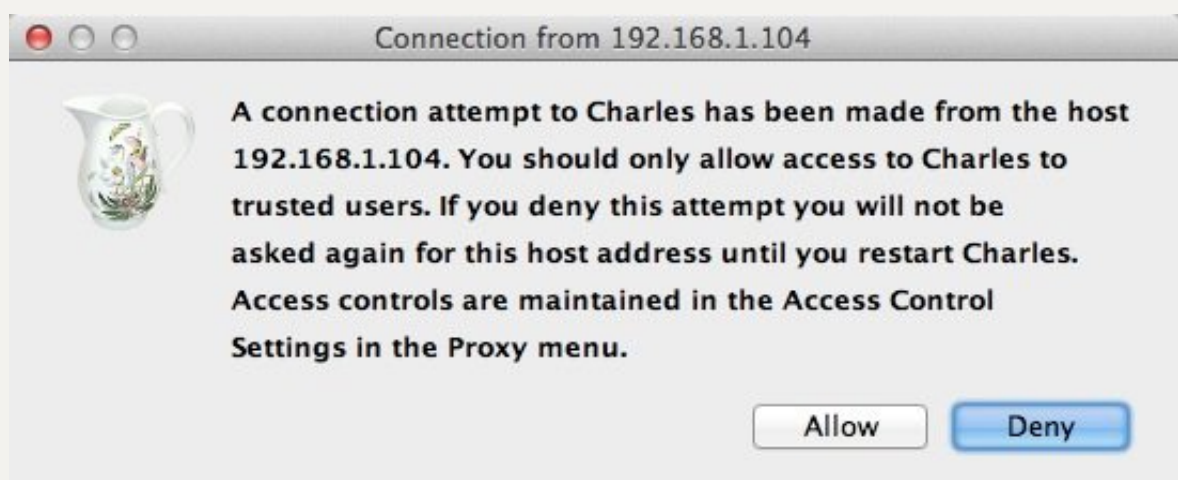
Charles 通常用来截取本地上的网络封包，但是当我们需要时，我们也可以用来截取其它设备上的网络请求。下面我就以 iPhone 为例，讲解如何进行相应操作。

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，并且勾上 `"Enable transparent HTTP proxying"` 就完成了在 Charles 上的设置。如下图所示：

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：



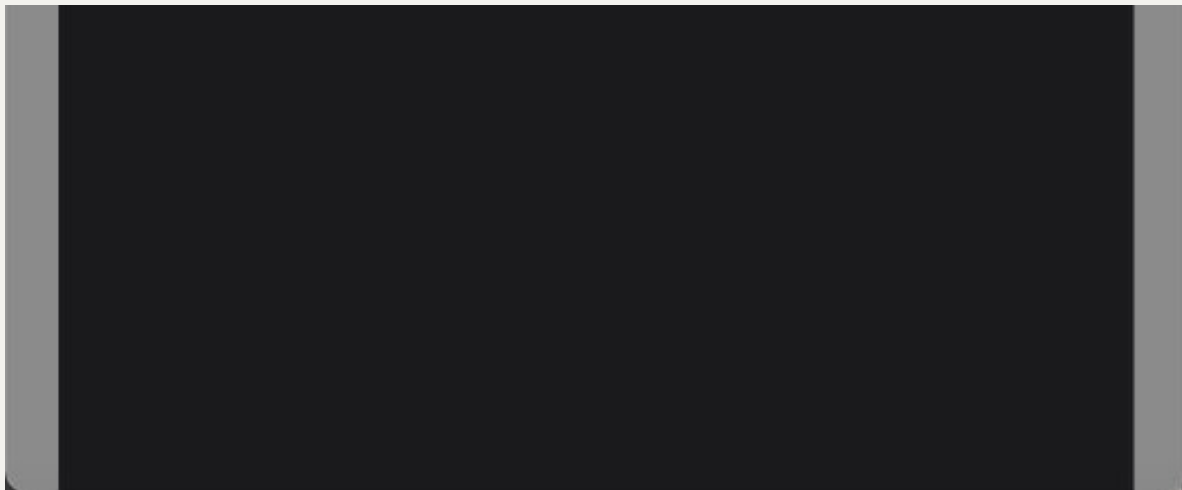
设置好之后，我们打开 iPhone 上的任意需要网络通讯的程序，就可以看到 Charles 弹出 iPhone 请求连接的确认菜单（如下图所示），点击“Allow”即可完成设置。



手机截取 **Https** 通讯信息:

手机网页输入 `chis.pro/ssl` 然后下载安装信任





这样子就可以抓取手机https请求啦

常见功能：

模拟慢速网络：

在做移动开发的时候，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：

Throttle Settings

☒ Enable Throttling

☐ Only for selected hosts

Location

Add Remove

Throttle preset: 56 kbps Modem

	Download	Upload
Bandwidth (kbps):	57.6	33.6
Utilisation (%):	70	70
Round-trip latency (ms):	250	
MTU (bytes):	576	
Reliability (%):	100	
Stability (%):	100	
Unstable quality range (%):	100	100

Add Preset Remove Preset

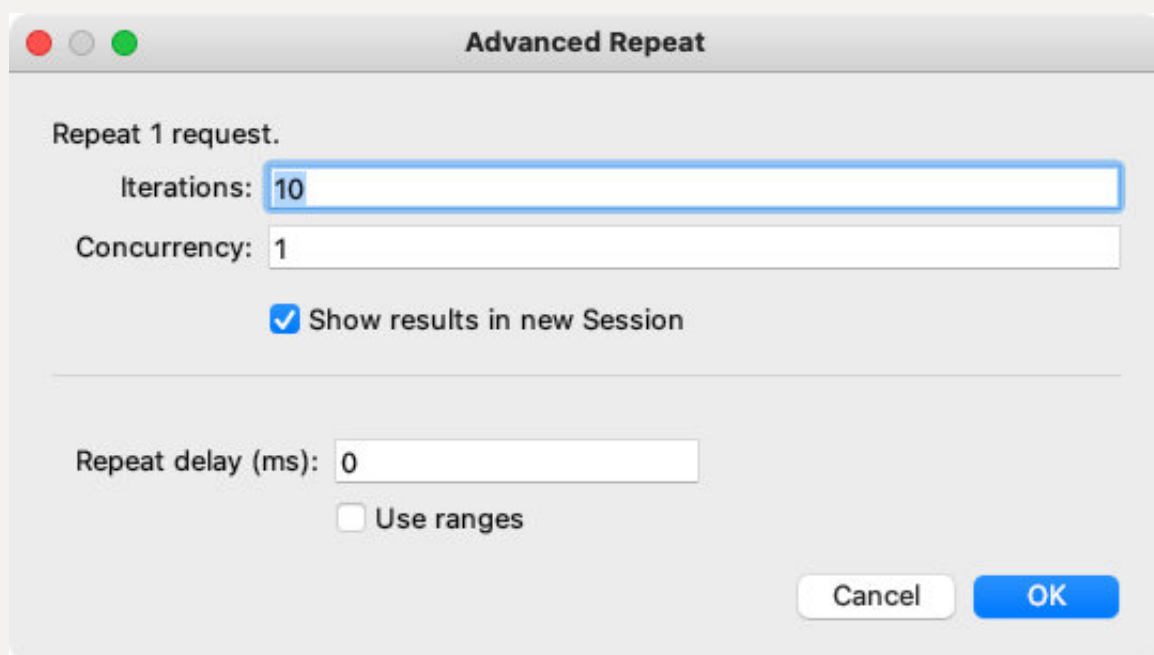
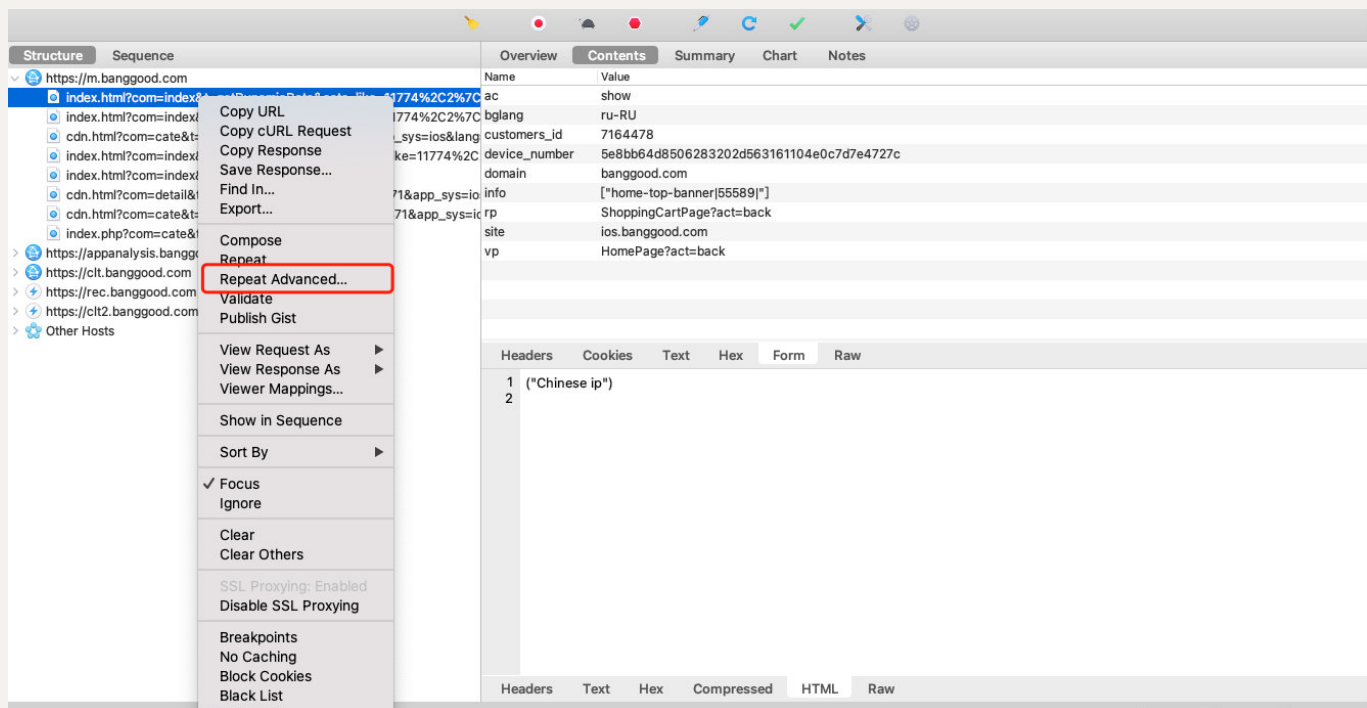
Import Export ? Cancel OK

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

给服务器做压力测试：

我们可以使用 Charles 的 Repeat 功能来简单地测试服务器的并发处理能力，方法如下。

我们在想打压的网络请求上（POST 或 GET 请求均可）右击，然后选择 Repeat Advanced 菜单项，如下所示：



点击ok,你就可以发起多次的网络请求了

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。

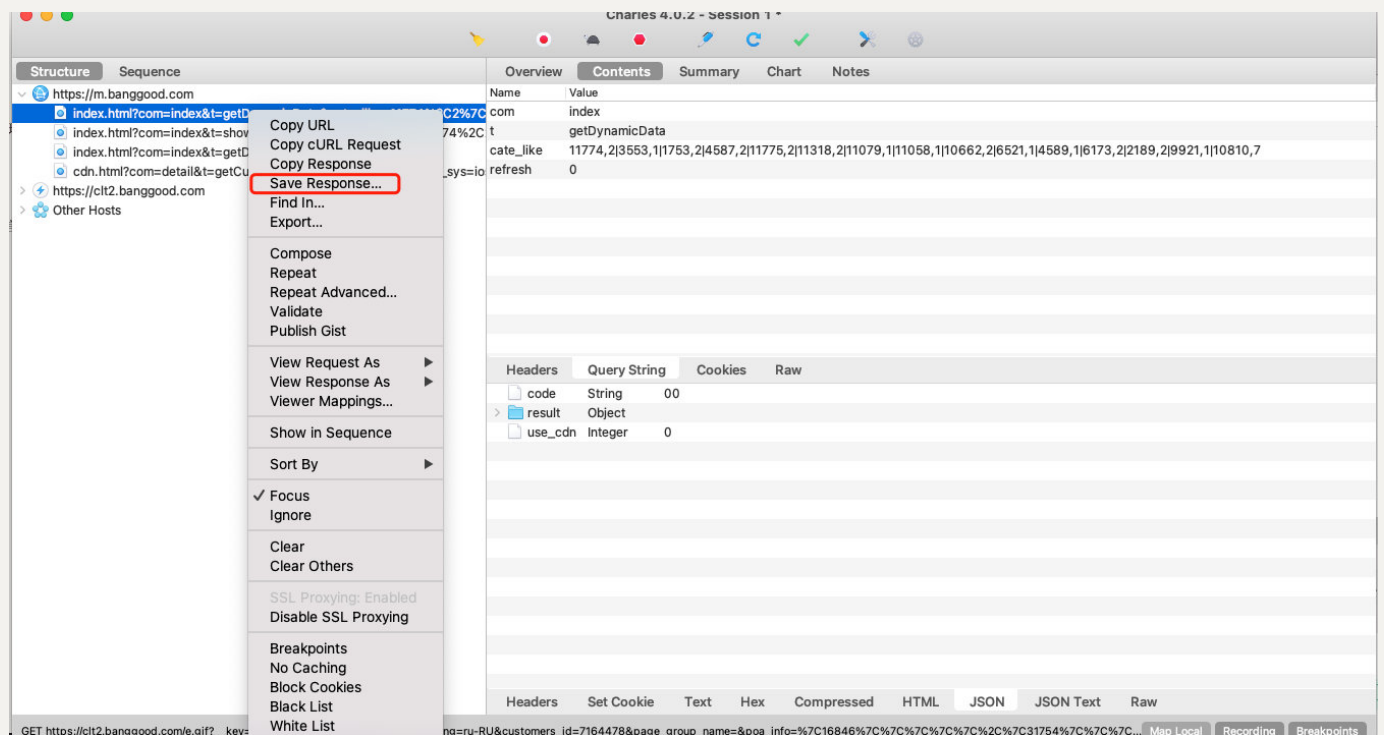
根据具体的需求，Charles 提供了 Map 功能、Rewrite 功能以及 Breakpoints 功能，都可以达到修改服务器返回内容的目的。这三者在功能上的差异是：

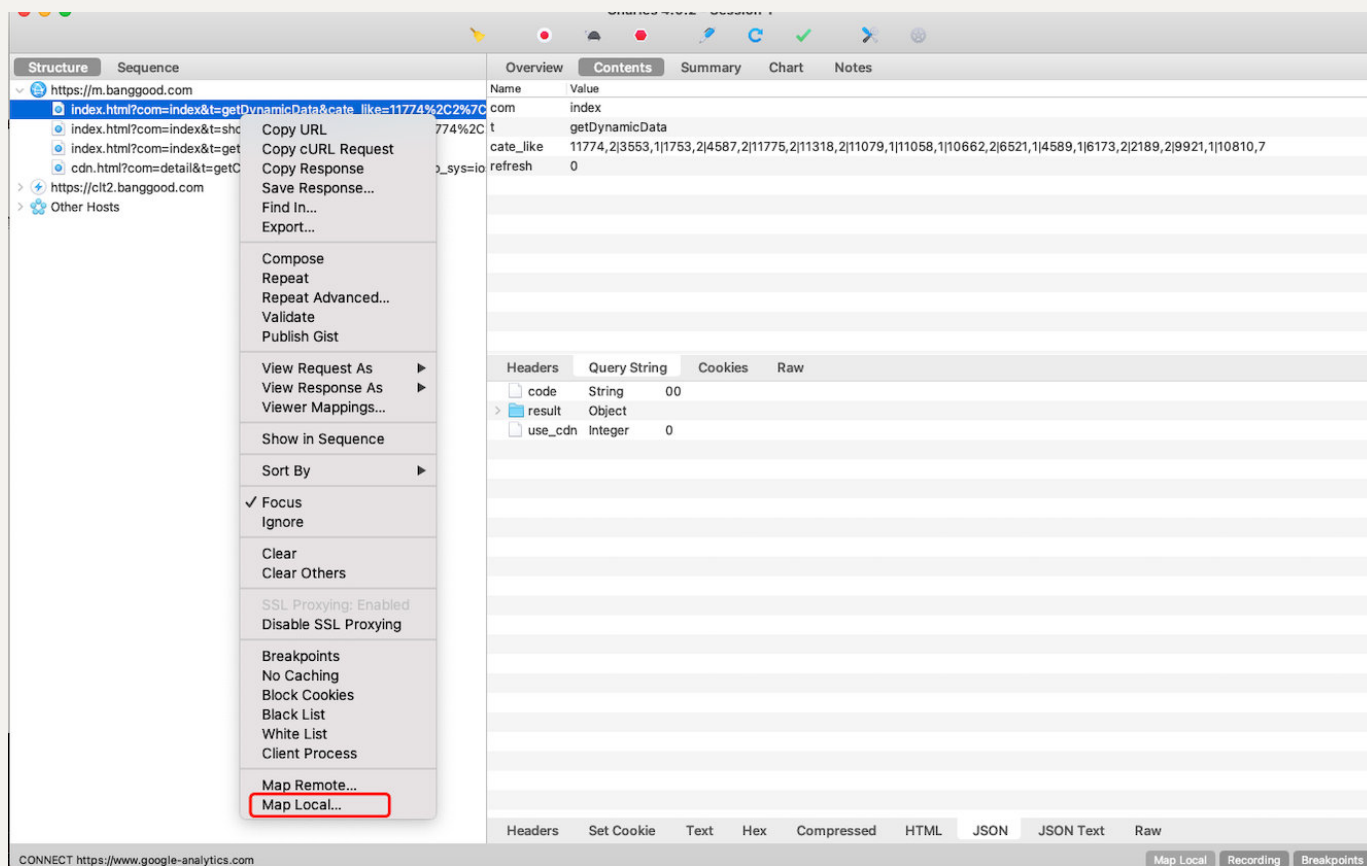
- Map 功能适合长期地将某一些请求重定向到另一个网络地址或本地文件。
- Rewrite 功能适合对网络请求进行一些正则替换。
- Breakpoints 功能适合做一些临时性的修改。

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

对于 Map Local,我们经常先保存一份数据,稍作修改,然后稍加修改，成为我们的目标映射文件。





Map Local 在使用的时候，有一个潜在的问题，就是其返回的 Http Response Header 与正常的请求并不一样。这个时候如果客户端校验了 Http Response Header 中的部分内容，就会使得该功能失效。解决办法是同时使用 Map Local 以下面提到的 Rewrite 功能，将相关的 Http 头 Rewrite 成我们希望的内容。

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。

The image shows a 'Rewrite Rule' dialog box with the following settings:

- Type:** Body
- Where:** ☐ Request, ☒ Response
- Match:**
 - Enter text to match or leave blank to match all.
 - Name:
 - Value:
 - ☐ Match whole value
 - ☐ Case sensitive
 - ☐ Regex
- Replace:**
 - Name:
 - Value:
 - ☐ Replace first, ☒ Replace all
 - Enter new values or leave blank for no change. If using regex matches you may enter references to groups, eg. \$1

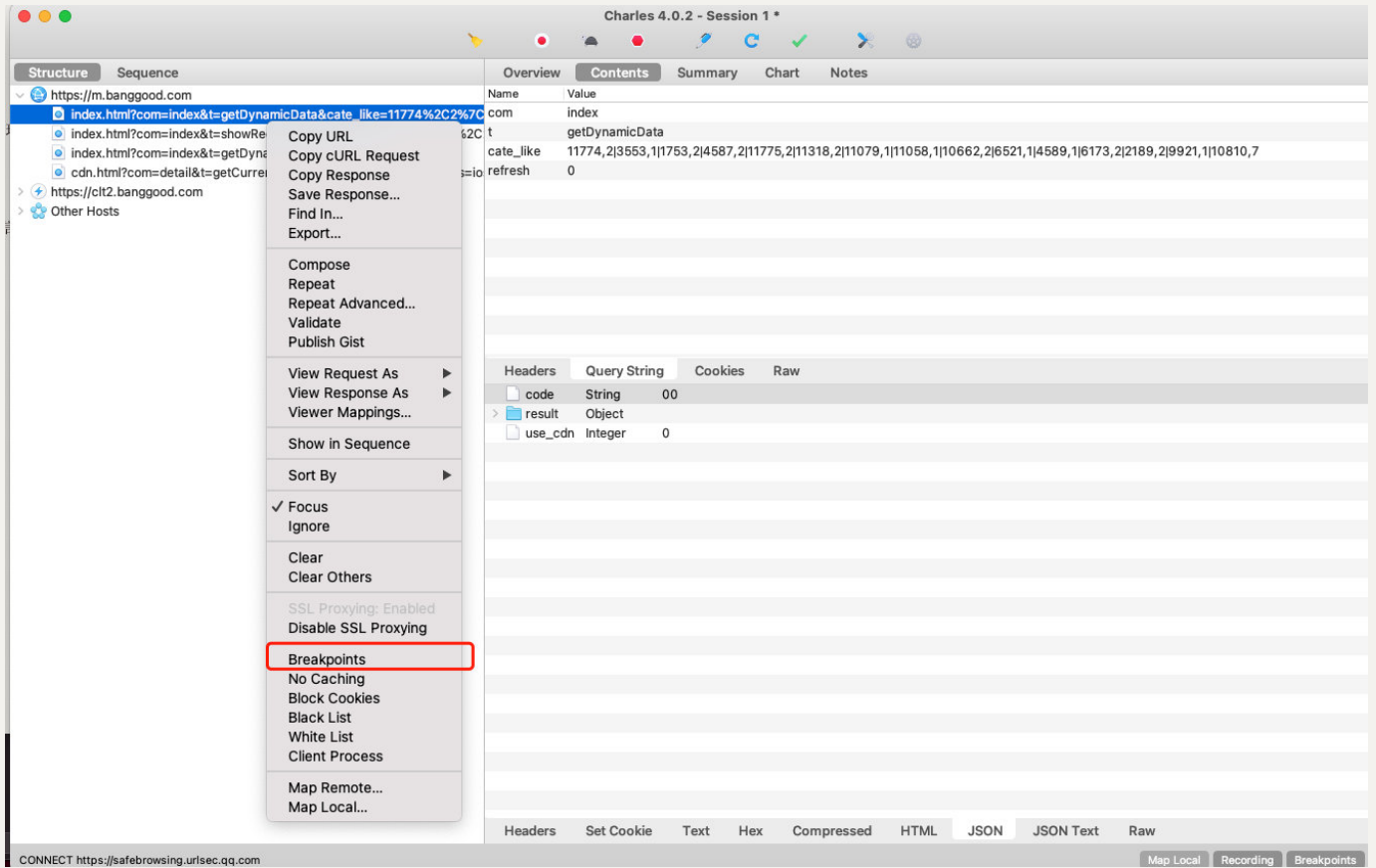
Buttons at the bottom: ? (help), Cancel, OK.

图中是将返回值的所有00改为01

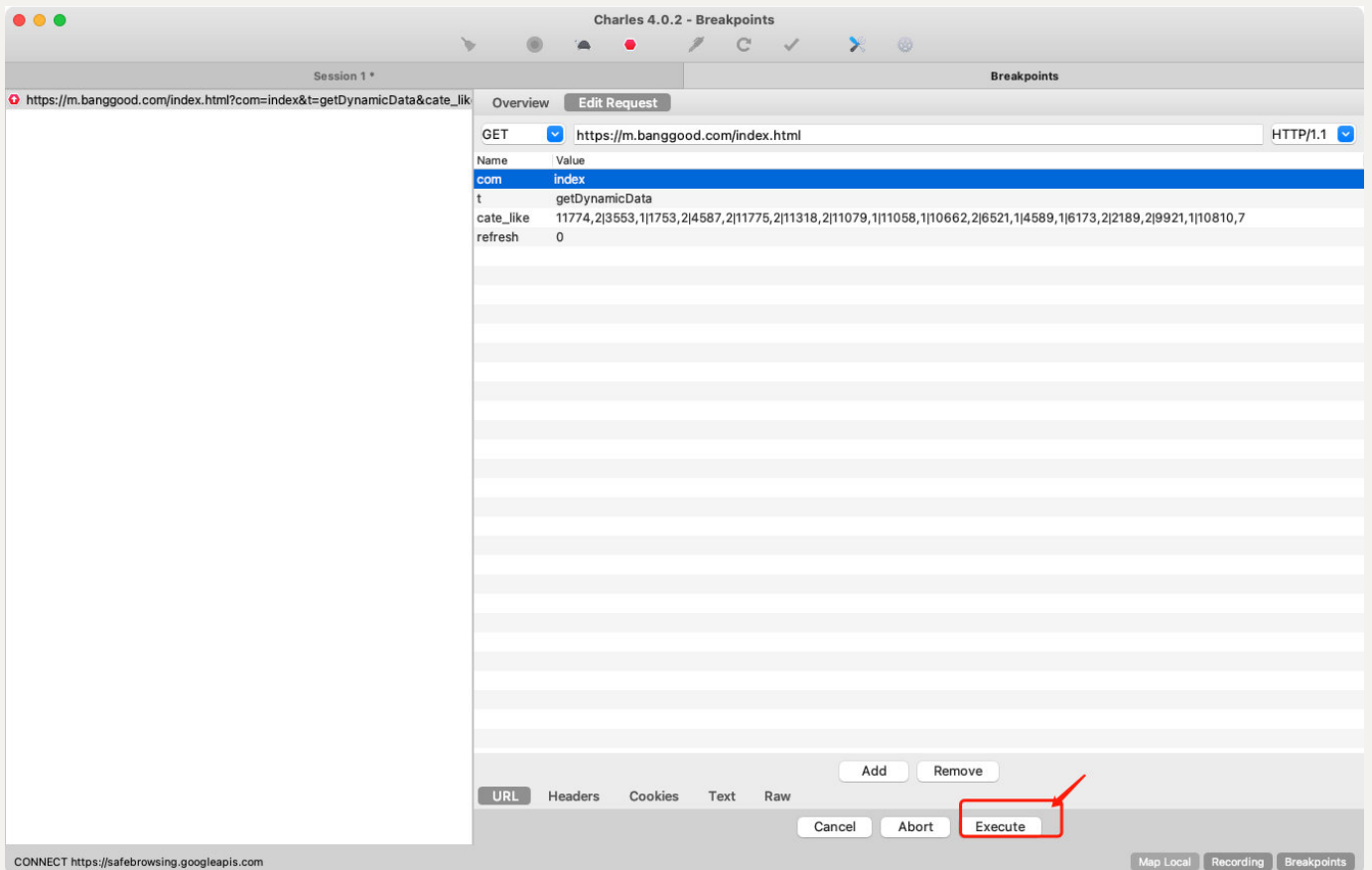
Breakpoints功能:

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。

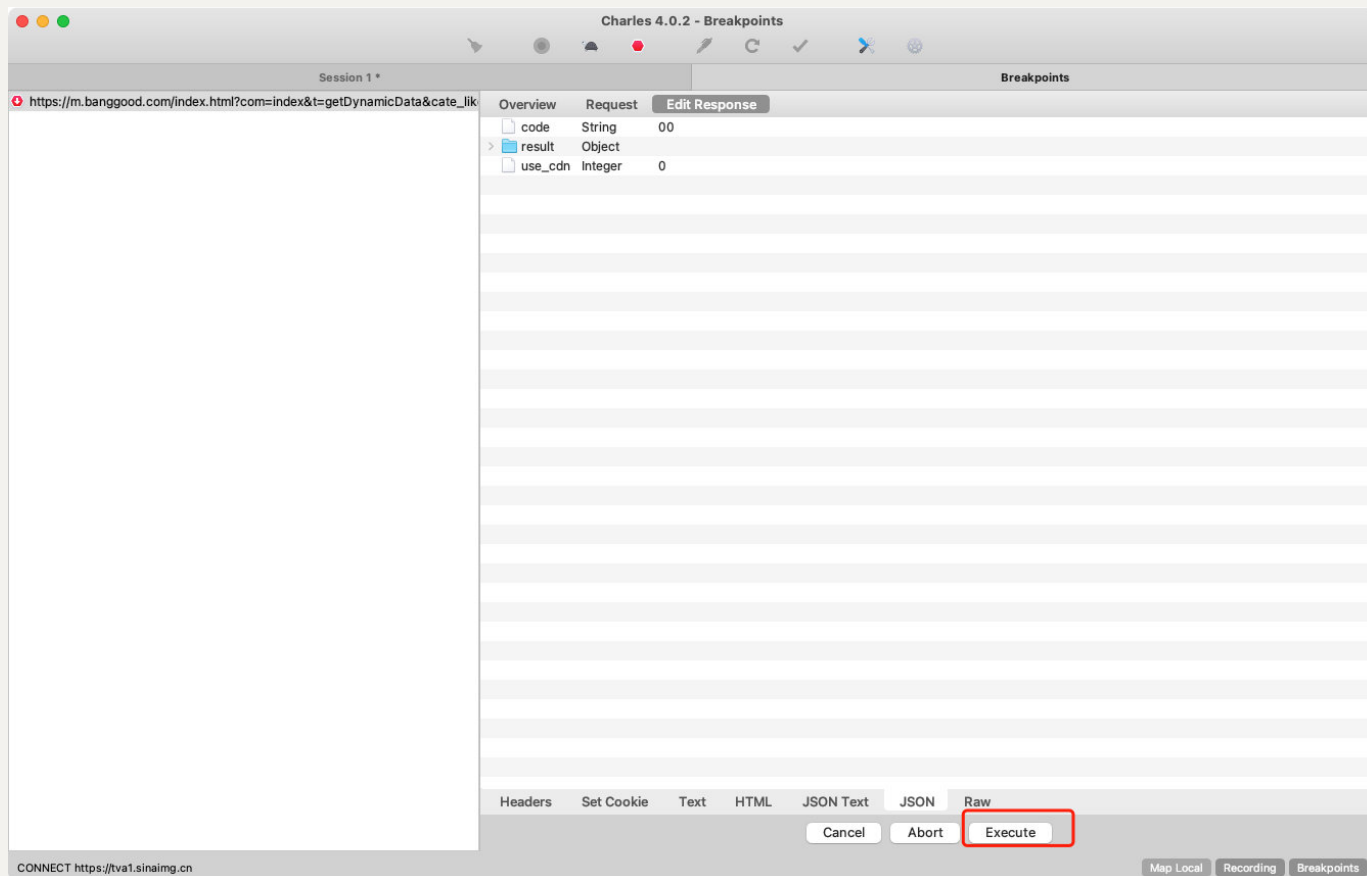
Breakpoints 功能类似我们在 Xcode 中设置的断点一样，当指定的网络请求发生时，Charles 会截获该请求，这个时候，我们可以在 Charles 中临时修改网络请求的返回内容。



再次请求后,就会断在发起网络前,让你修改参数。还有断在网络请求后,给你修改返回值这里



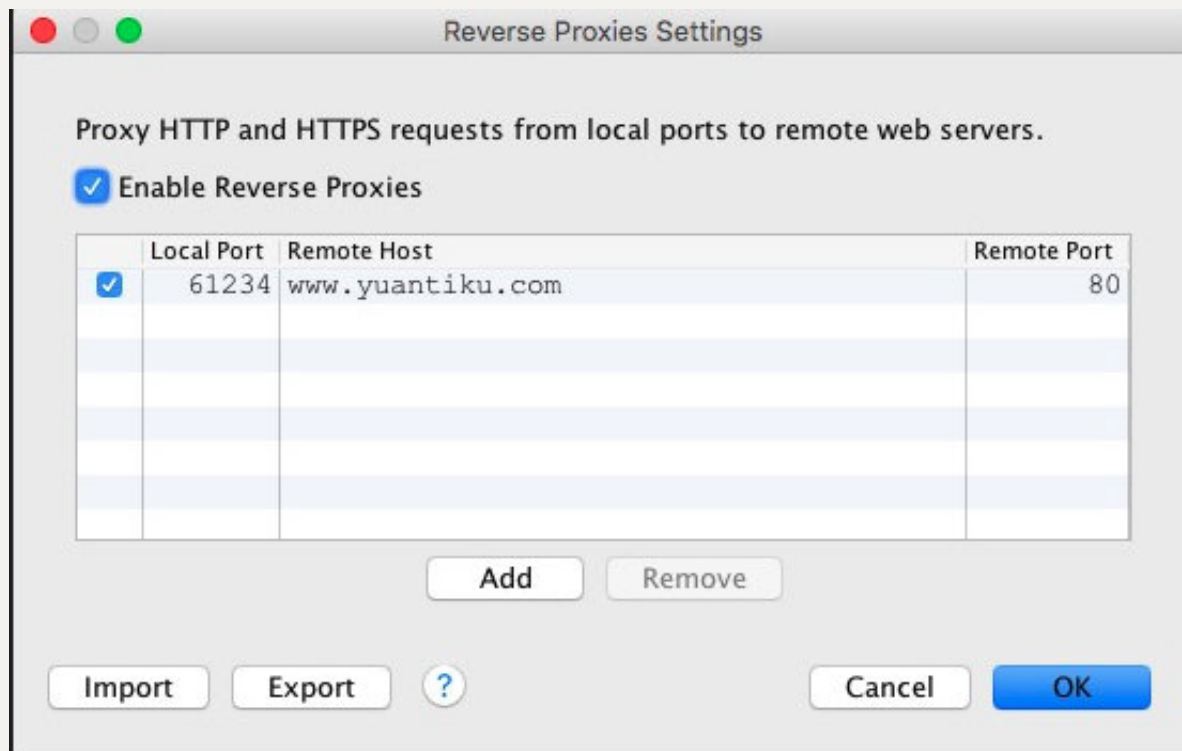
按Exexute,会发起网络请求,返回结果断点



返回结果断点,让你修改返回值。

设置外部代理:

Charles 的反向代理功能允许我们将本地的端口映射到远程的另一个端口上。例如，在下图中，我将本机的 61234 端口映射到了远程（www.yuantiku.com）的80端口上了。这样，当我访问本地的80-gp7i66a966a1m8fsfk.xn--%2C-fi0bx3wi0cz0ixtbm64chv1b4yh9yp/) 61234 端口时，实际返回的内容会由 www.yuantiku.com 的 80 端口提供。



设置外部代理，解决与翻墙软件的冲突：

Charles 的原理是把自己设置成系统的代理服务器，但是在中国，由于工作需要，我们常常需要使用 Google 搜索，所以大部分程序员都有自己的翻墙软件，而这些软件的基本原理，也是把自己设置成系统的代理服务器，来做到透明的翻墙。

为了使得两者能够和平共处，我们可以在 Charles 的 **External Proxy Settings** 中，设置翻墙的代理端口以及相关信息。同时，我们也要关闭相关翻墙软件的自动设置，使其不主动修改系统代理，避免 Charles 失效

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