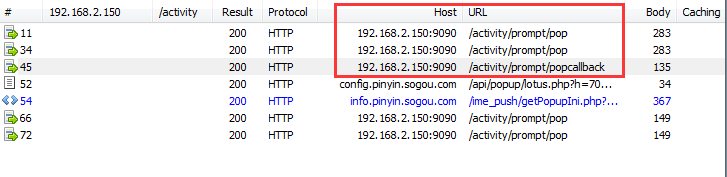
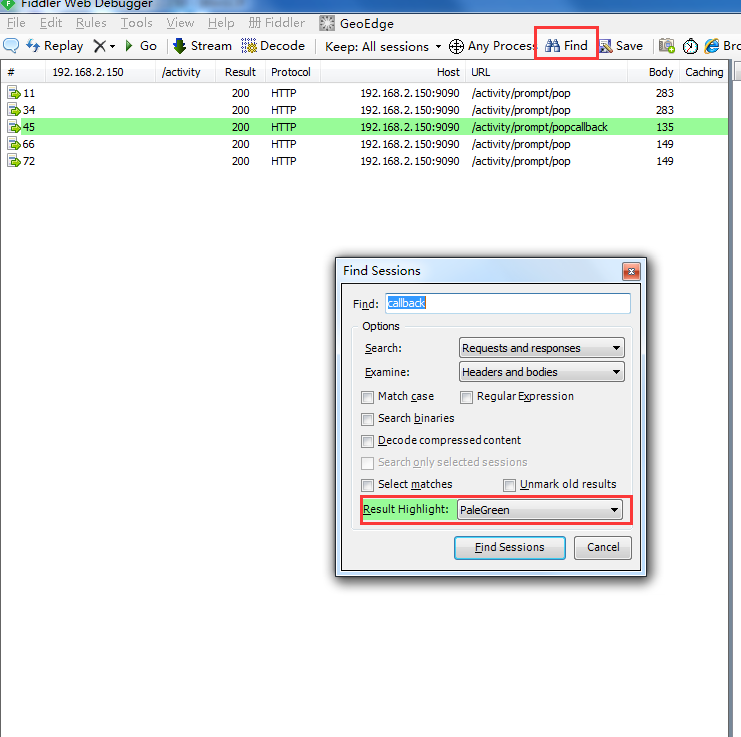
Fiddler是一款比较好的监控工具，它能够记录客户端和服务器之间的所有 HTTP请求，用来测试手机App的功能也很强大。关于Fiddler的如何安装和使用在珍丹的PPT中有介绍了。

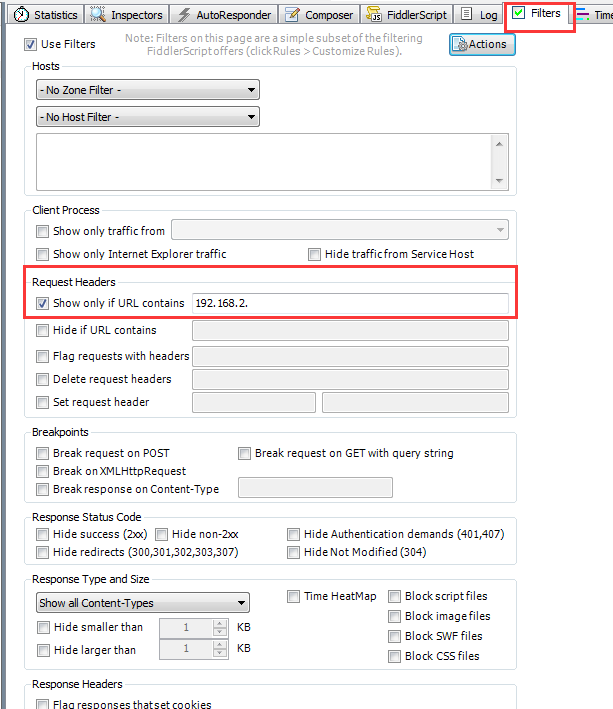
分享几个自己在使用中用到的功能：

1. 通过Host和URL可以很清楚的看到自己当前访问的服务器host，端口号以及对应的接口方法。在排查问题和验证配置是否正确时十分便捷。



1. 简单的filter和find功能使用。在Filter中输入自己的过滤条件，可以过滤出只对自己有用的请求。Find功能，可以进一步定位自己想要看的那条记录。



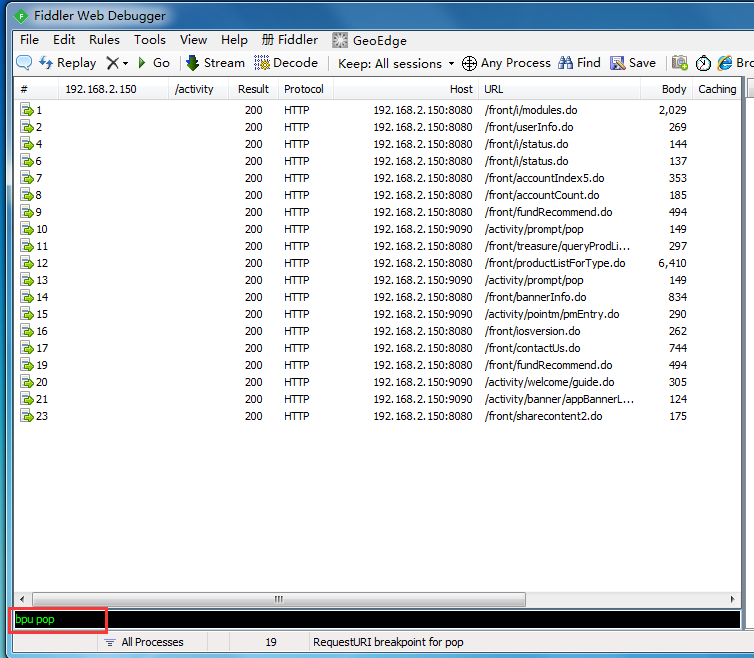


1. Fiddler中设置断点修改Request

实例：在测试红包弹窗时，红包弹窗时根据设备号来决定是否弹窗，如果没有多余的设备，可以通过手动改request请求来测试红包的弹窗功能。

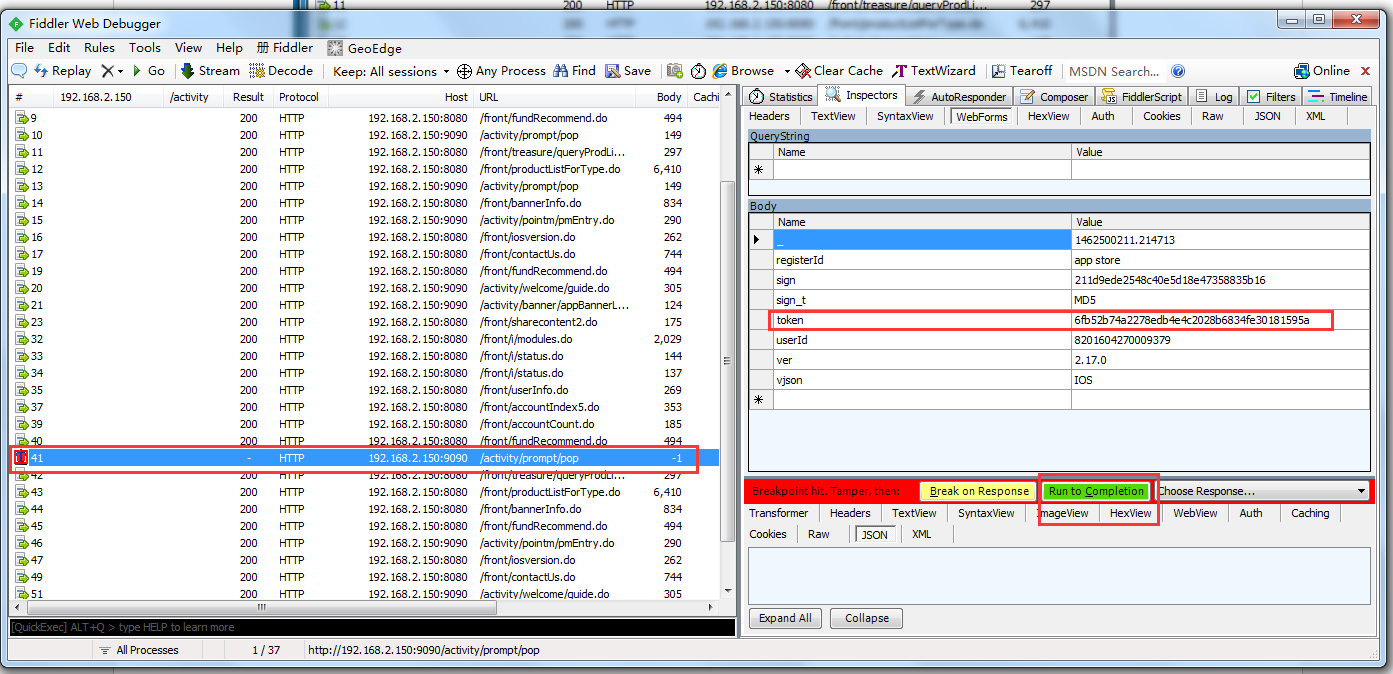
首先输入你需要设置断点的request，输入关键字即可。

在命令行中输入命令：bpu pop（pop为弹窗请求的关键字）



这时候所有的pop请求都会被阻断。这时候去更改token的值，然后点run to completion，

就能看到对应的request被修改，弹窗显示在App端。

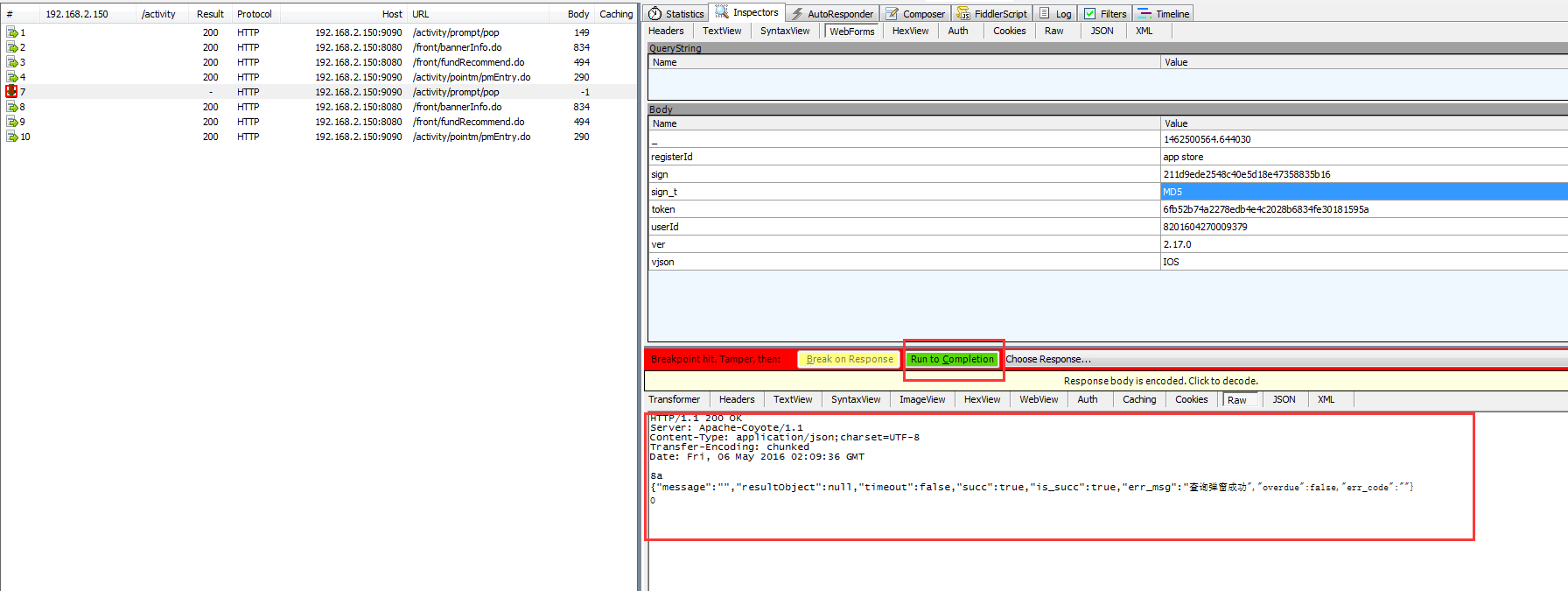


如何消除命令呢？在命令行中输入命令 bpu

1. Fiddler中设置断点修改Response

类似3. 在命令行中输入命令：bpafter pop

手动修改response中的内容，然后点击run to completion。也可以达到让App弹窗的效果。

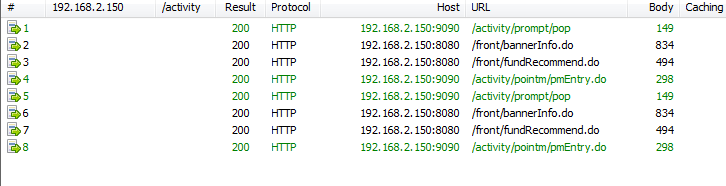
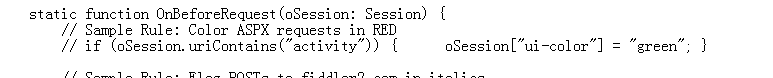


如何消除命令呢？  在命令行中输入命令 bpafter

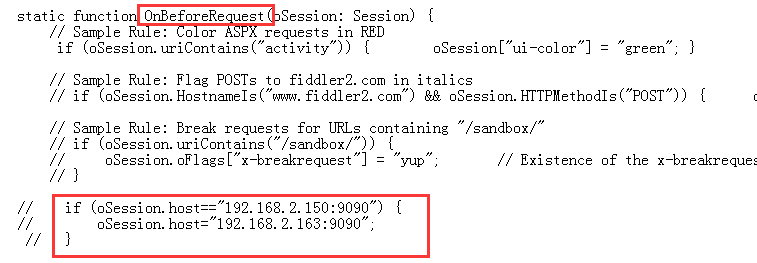
1. FiddlerScript中的一些用法。

在操作界面点击Ctrl+R，会弹出FiddlerScript，如果没有则会引导到安装的网页完成安装即可。通过对FiddlerScript的修改可以帮助我们更好的测试。

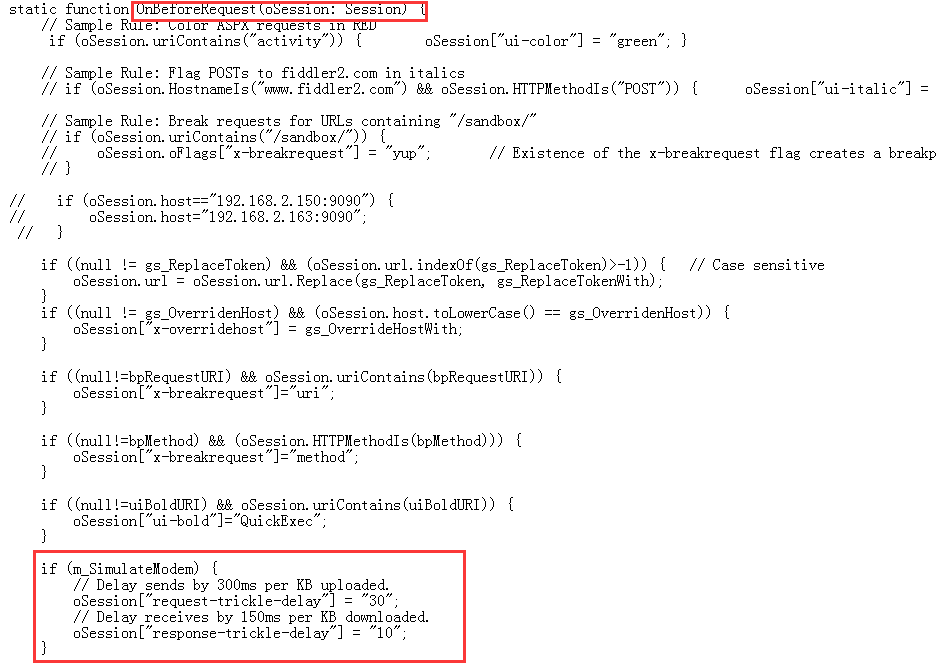
1. 更改需要查看的请求的颜色。在OnBeforeRequest中，设置对应的颜色。下图中就设置activity所有系统为绿色，方便查看。

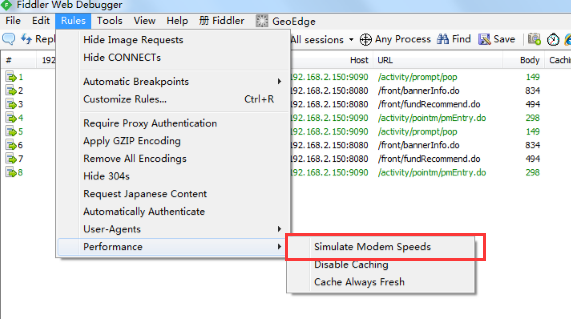


1. 更改需要访问的host地址。在OnBeforeRequest中，设置对应的DNS转化。比如你正在150的服务器上测试新的代码，但是想看看163上的老代码的现象，或者有些配置文件不对，你可以通过这种方式去把所有调150的请求转化为去调163的请求。



1. 设置对应的网络延迟。还是以测试首页弹窗为例，在首页弹窗下载资源过程中如果切换到资产或者我的页面，就不该再弹窗。但是网络好或者手速不快的情况下很难实现这个点的测试。在OnBeforeRequest中，设置对应的网络延迟，如下图。下图中把下载速度限制到每10毫秒下载1KB，对于500K左右的图片，就需要5S左右的时间去下载。设置好以后需要手动激活延迟功能，测试完再取消掉。





1. Composer的简单使用。在fiddler中，还支持自己生成request。选择composer，然后自定义request或者直接从当前的请求中随便选中一条拖曳。然后更改对应的参数，点击excute，就可以很方便的做部分接口的测试。

