**【HTTP】Fiddler（一） - Fiddler简介**

**1.为什么是Fiddler?**

抓包工具有很多，小到最常用的web调试工具firebug，大到通用的强大的抓包工具wireshark.为什么使用fiddler?原因如下：

a.Firebug虽然可以抓包，但是对于分析http请求的详细信息，不够强大。模拟http请求的功能也不够，且firebug常常是需要“无刷新修改”，如果刷新了页面，所有的修改都不会保存。

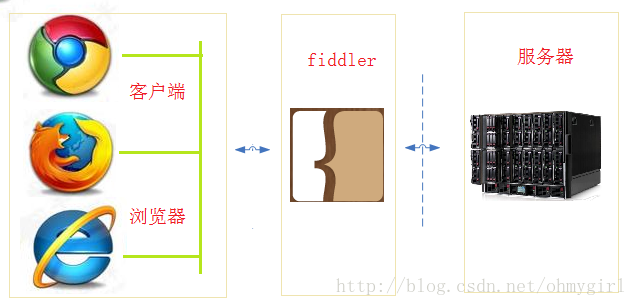
b.Wireshark是通用的抓包工具，但是比较庞大，对于只需要抓取http请求的应用来说，似乎有些大材小用。

c.Httpwatch也是比较常用的http抓包工具，但是只支持IE和firefox浏览器（其他浏览器可能会有相应的插件），对于想要调试chrome浏览器的http请求，似乎稍显无力，而Fiddler2 是一个使用本地 127.0.0.1:8888 的 HTTP 代理，任何能够设置 HTTP 代理为 127.0.0.1:8888 的浏览器和应用程序都可以使用 Fiddler。

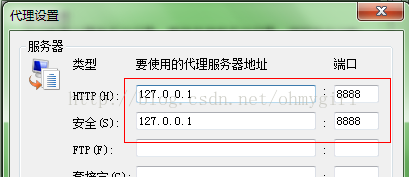
**2.什么是Fiddler?**

Fiddler是位于客户端和服务器端的HTTP代理，也是目前最常用的http抓包工具之一 。 它能够记录客户端和服务器之间的所有 HTTP请求，可以针对特定的HTTP请求，分析请求数据、设置断点、调试web应用、修改请求的数据，甚至可以修改服务器返回的数据，功能非常强大，是web调试的利器。

既然是代理，也就是说：客户端的所有请求都要先经过Fiddler，然后转发到相应的服务器，反之，服务器端的所有响应，也都会先经过Fiddler然后发送到客户端，基于这个原因，Fiddler支持所有可以设置http代理为127.0.0.1:8888的浏览器和应用程序。使用了Fiddler之后，web客户端和服务器的请求如下所示：

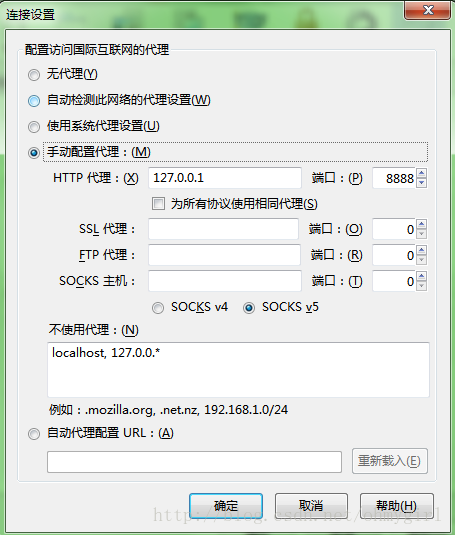


Fiddler 作为系统代理，当启用 Fiddler 时，IE 的PROXY 设定会变成 127.0.0.1:8888，因此如果你的浏览器在开启fiddler之后没有设置相应的代理，则fiddler是无法捕获到HTTP请求的。如下是启动Fiddler之后，IE浏览器的代理设置：



以Firefox为例，默认情况下，firefox是没有启用代理的（如果你安装了proxy等代理工具或插件，是另外一种情况），在firefox中配置http代理的步骤如下：

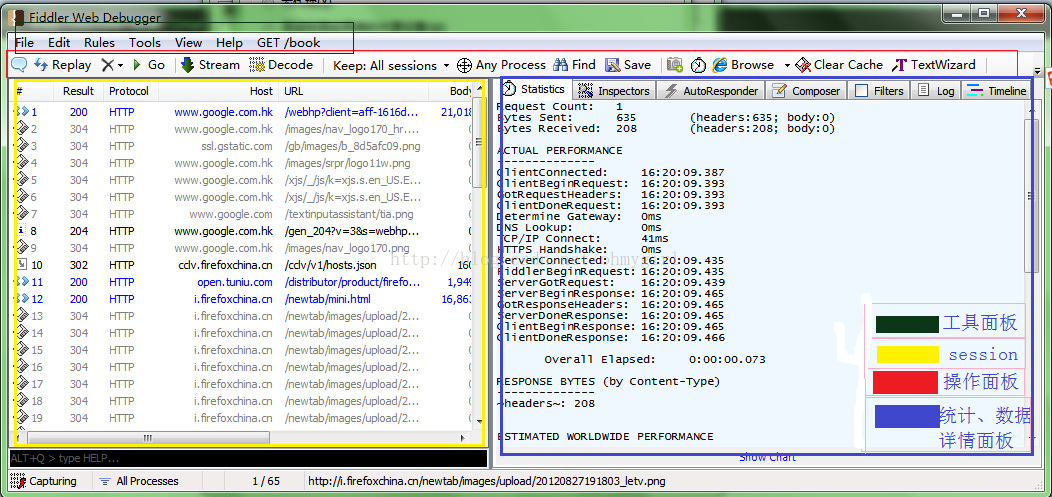
工具->选项->高级->网络->设置  。并配置相应的代理如下：



就可以使用Fiddler抓取Firefox的HTTP请求了。

**3.Fiddler使用界面简介**

Fiddler主界面的布局如下：

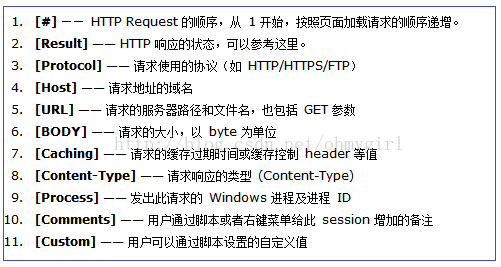


主界面中主要包括四个常用的块：

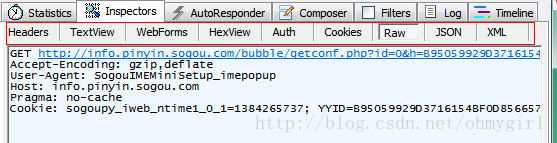
1.Fiddler的菜单栏，上图绿色部分。包括捕获http请求，停止捕获请求，保存http请求，载入本地session、设置捕获规则等功能。

2.Fiddler的工具栏,上图红色部分。包括Fiddler针对当前view的操作（暂停，清除session,decode模式、清除缓存等）。

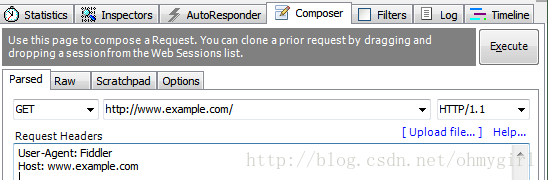
3.web Session面板，上图黄色区域，主要是Fiddler抓取到的每条http请求（每一条称为一个session）,主要包含了请求的url，协议，状态码，body等信息，详细的字段含义如下图所示：



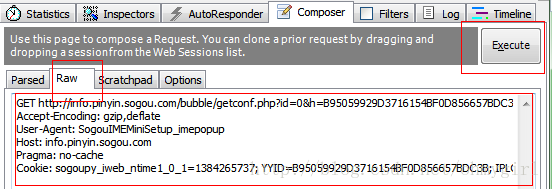
4.详情和数据统计面板。针对每条http请求的具体统计（例如发送/接受字节数，发送/接收时间，还有粗略统计世界各地访问该服务器所花费的时间）和数据包分析。如inspector面板下，提供headers、textview、hexview,Raw等多种方式查看单条http请求的请求报文的信息：



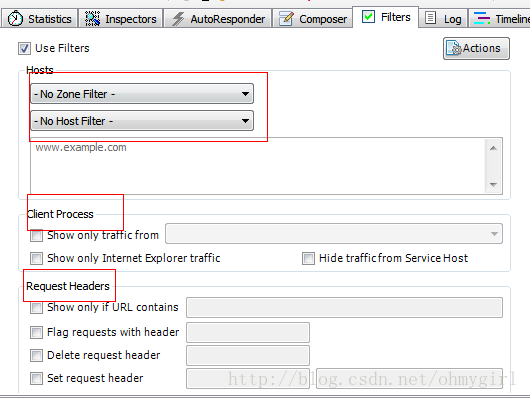
而composer面板下，则可以模拟向相应的服务器发送数据的过程（不错，这就是灌水机器人的基本原理,也可以是部分http flood的一种方式）。



也可以粘贴一次请求的raw http headers,达到模拟请求的目的：



Filter标签则可以设置Fiddler的过滤规则，来达到过滤http请求的目的。最简单如：过滤内网http请求而只抓取internet的http请求，或则过滤相应域名的http请求。Fiddler的过滤器非常强大，可以过滤特定http状态码的请求，可以过滤特定请求类型的http请求（如css请求，image请求，js请求等），可以过滤请求报文大于或则小于指定大小（byte）的请求：



请多的过滤器规则需要一步一步去挖掘。