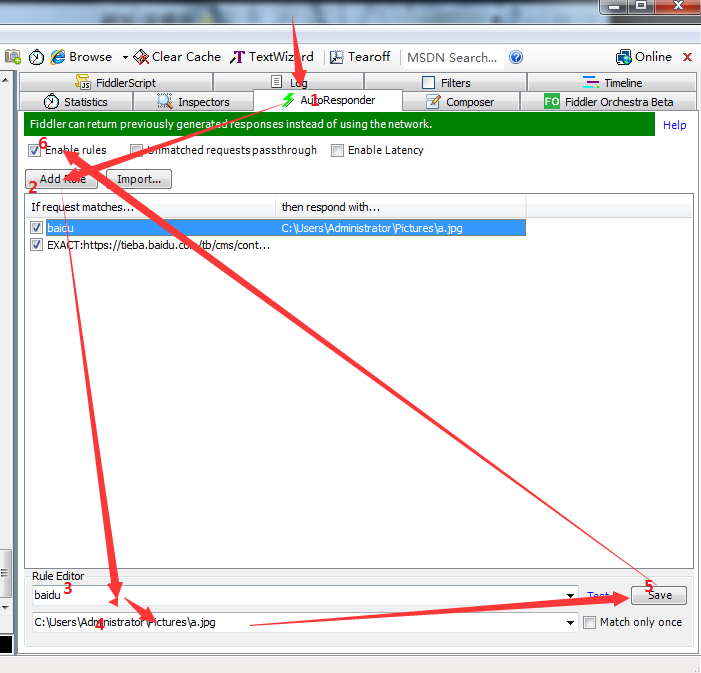
Fiddler抓包：Fiddler是通过改写HTTP代理，让数据从它那通过，来监控并且截取到数据

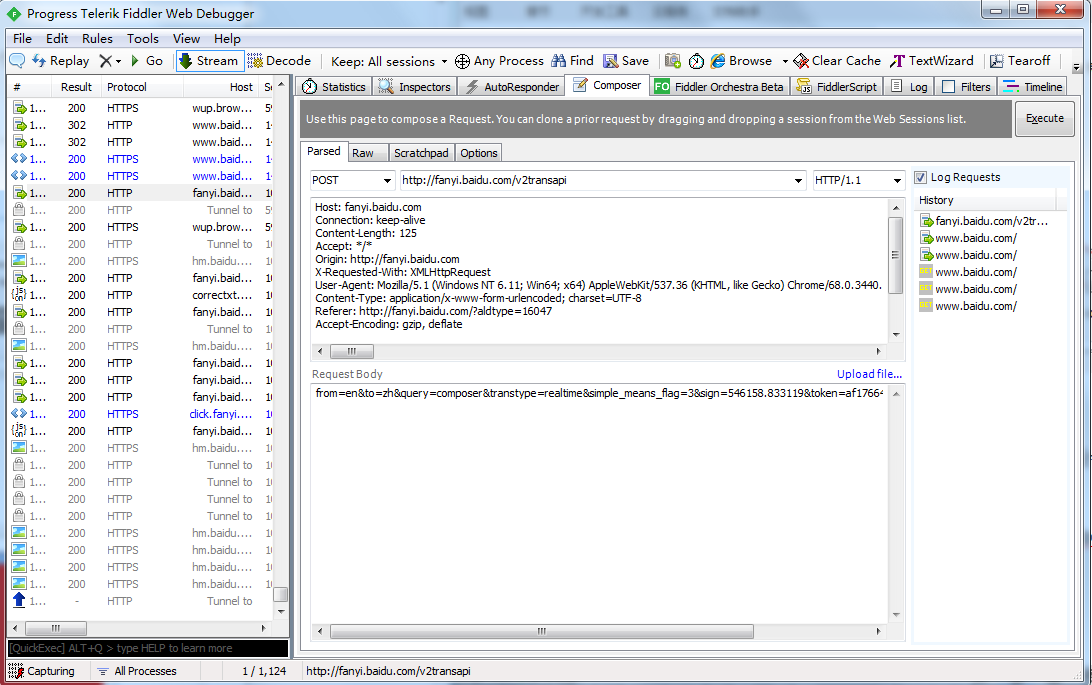
#### **. AutoResponder 允许拦截指定规则的请求**



#### **Composer 自定义请求发送服务器**

Composer允许自定义请求发送到服务器，可以手动创建一个新的请求，也可以在会话表中，拖拽一个现有的请求

Parsed模式下你只需要提供简单的URLS地址即可（如下图，也可以在RequestBody定制一些属性，如模拟浏览器User-Agent）



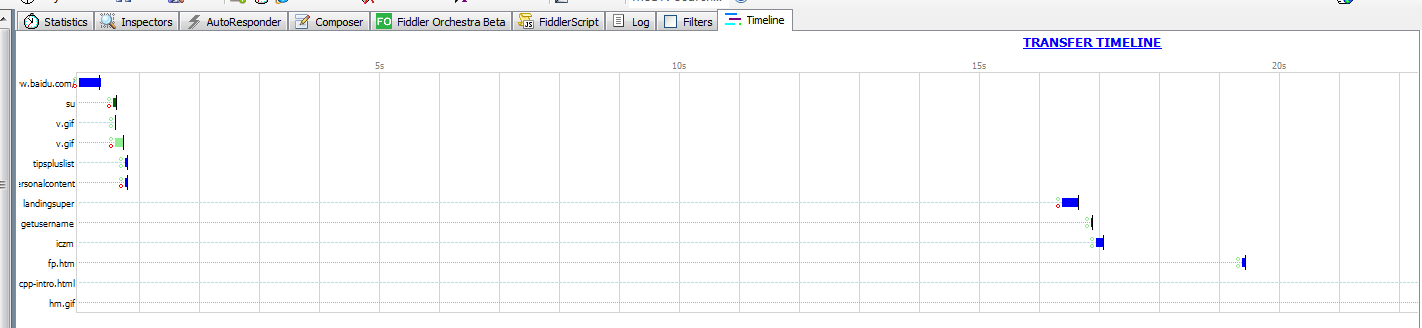
#### **. Filters 请求过滤规则**

Fiters 是过滤请求用的，左边的窗口不断的更新，当你想看你系统的请求的时候，你刷新一下浏览器，一大片不知道哪来请求，看着碍眼，它还一直刷新你的屏幕。这个时候通过过滤规则来过滤掉那些不想看到的请求。

#### 

#### **. Timeline 请求响应时间**

在左侧会话窗口点击一个或多个（同时按下 Ctrl 键），Timeline 便会显示指定内容从服务端传输到客户端的时间：



### **2. Fiddler 设置解密HTTPS的网络数据**

Fiddler可以通过伪造CA证书来欺骗浏览器和服务器。大概原理就是在浏览器面前Fiddler伪装成一个HTTPS服务器，而在真正的HTTPS服务器面前Fiddler又装成浏览器，从而实现解密HTTPS数据包的目的。

命令：

#### **2.1 基本命令**

|  |  |  |  |
| --- | --- | --- | --- |
| ****命令**** | ****对应请求项**** | ****介绍**** | ****示例**** |
| ? | All | 问号后边跟一个字符串，可以匹配出包含这个字符串的请求 | ?google |
| > | Body | 大于号后面跟一个数字，可以匹配出请求大小，大于这个数字请求 | >1000 |
| < | Body | 小于号跟大于号相反，匹配出请求大小，小于这个数字的请求 | <100 |
| = | Result | 等于号后面跟数字，可以匹配HTTP返回码 | =200 |
| @ | Host | @后面跟Host，可以匹配域名 | @www.baidu.com |
| select | Content-Type | select后面跟响应类型，可以匹配到相关的类型 | select image |
| cls | All | 清空当前所有请求 | cls |
| dump | All | 将所有请求打包成saz压缩包，保存到“我的文档Fiddler2Captures”目录下 | dump |
| start | All | 开始监听请求 | start |
| stop | All | 停止监听请求 | stop |

#### **2.2 断点命令（进行断点测试）**

|  |  |  |  |
| --- | --- | --- | --- |
| ****命令**** | ****对应请求项**** | ****介绍**** | ****示例**** |
| bpafter | All | bpafter后边跟一个字符串，表示中断所有包含该字符串的请求 | bpafter baidu（输入bpafter解除断点） |
| bpu | All | 跟bpafter差不多，只不过这个是收到请求了，中断响应 | bpu baidu（输入bpu解除断点） |
| bps | Result | 后面跟状态吗，表示中断所有是这个状态码的请求 | bps 200（输入bps解除断点） |
| bpv / bpm | HTTP方法 | 只中断HTTP方法的命令，HTTP方法如POST、GET | bpv get（输入bpv解除断点） |
| g / Go | All | 放行所有中断下来的请求 | g |