

Fiddler 是啥？

百度百科里是这样介绍它的 - “Fiddler 是一个 web 调试代理。它能够记录所有[客户端](#)和服务端间的 http 请求，允许你监视，设置[断点](#)，甚至修改输入输出数据，fiddler 包含了一个强大的基于事件[脚本](#)的子系统，并且能够使用[.net 框架](#)语言扩展。”

所以无论你是从事什么开发，哪种语言，只要你想了解 HTTP，这个工具就值得你去了解，而且更重要的一点，这个工具是免费的。

Fiddler 就是以代理服务器的方式，监听系统的网络数据流动。

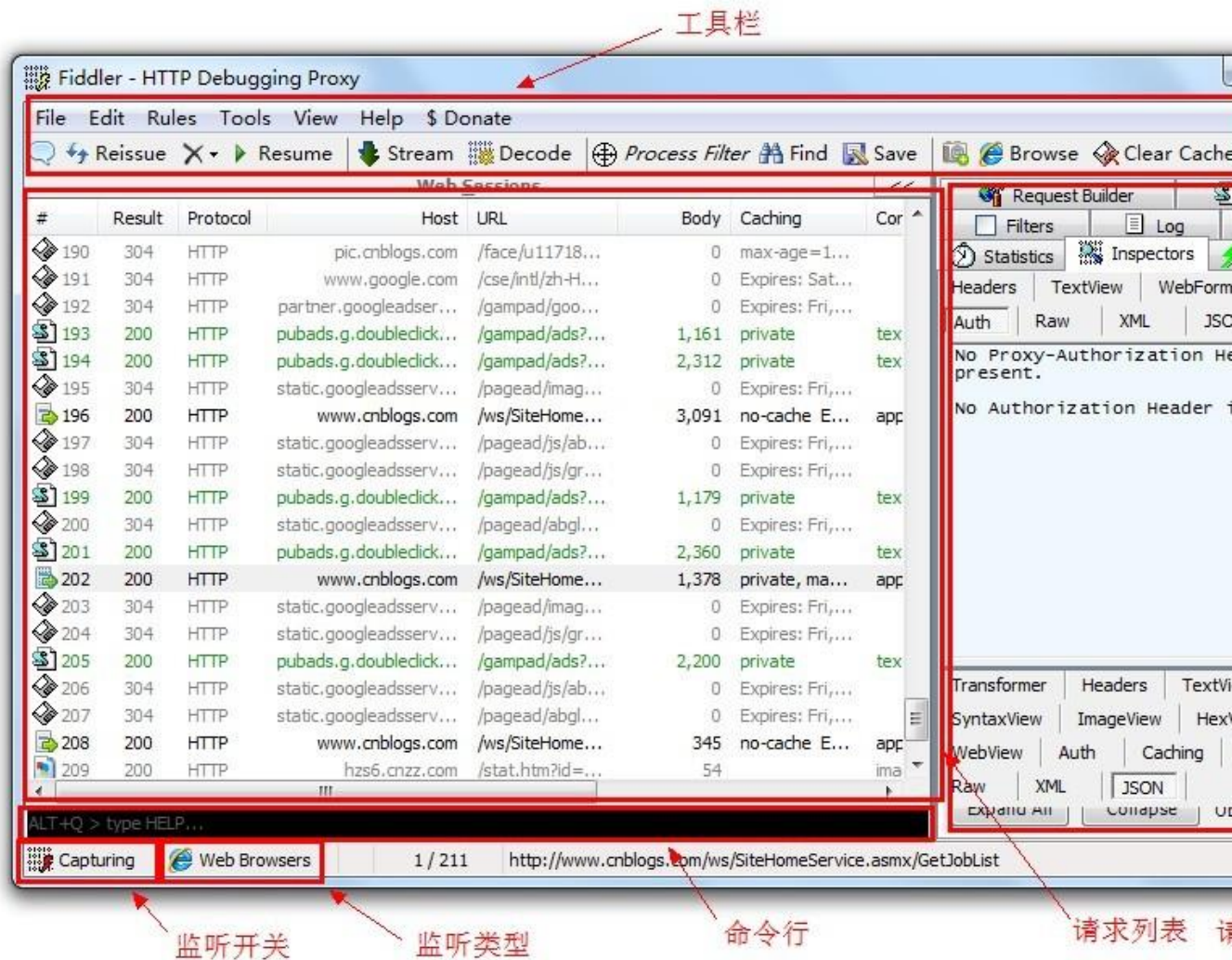
启动 Fiddler 后，所发生的网络数据流通过 Fiddler 进行中转，就可以看到 HTTP/HTTPS 数据流的信息，我们就可以通过对这些信息进行分析。Fiddler 还提供了清除 IE 缓存、请求构造器、文本转换工具等一系列工具，对前端开发工作很有价值。

Fiddler 的安装与下载：

Fiddler 下载地址：<http://www.fiddler2.com/fiddler2/>

假如你是早期的 XP 版本的系统在安装的过程中会提示你下载 .net framework 2.0 或以上版本。安装过程很简单，就不介绍了。

Fiddler 的使用界面和功能介绍：



监听开关 - 只有两种状态，用的时候就开着，不用就让丫休息。capturing 表示捕捉状态

监听类型 - 四种状态分别对应 监听所有请求；监听浏览器请求，监听非浏览器请求，和全部隐藏(Hide All)

命令行 - 就不作介绍了，难者不会，会者不难。我就属于前者，悲剧呀...

请求列表 - 请求列表的信息分别有 结果 (Result)，协议 (Protocol)，主机名 (Host)，网页地址 (URL)，内容大小 (Body)，缓存 (Caching)，响应的 HTTP 内容类型 (Content-Type)，请求所运行的程序 (Process)，注释 (Comments)，自定义 (Custom)

请求相关信息 - 右边这一大片都是数据流的相关信息的查看器，这些查看器提供很多查看形式，可以查看数据流的内容。

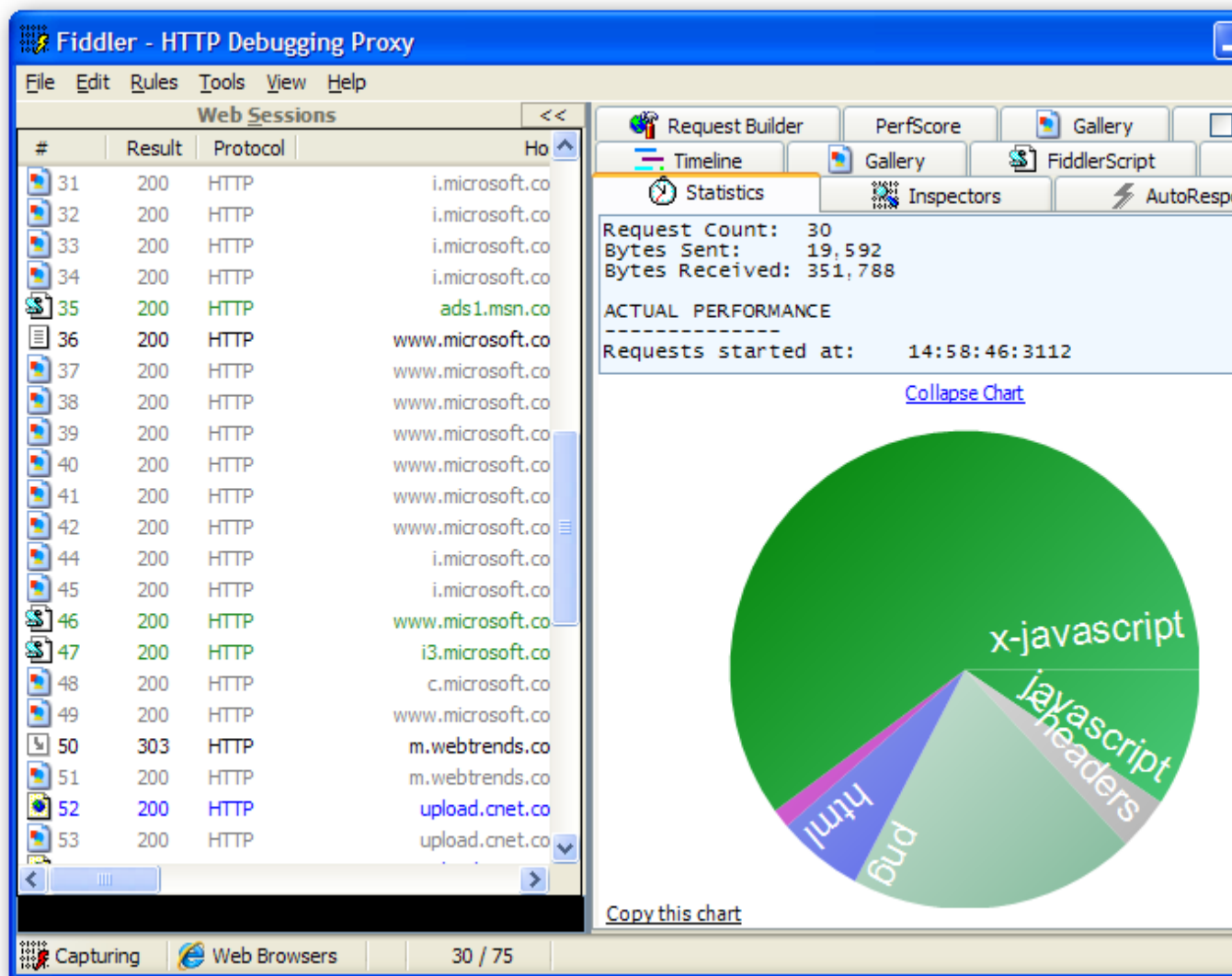
Fiddler 请求列表的 icon 对应具体的数据类型和状态，其含义是：

-  正在将请求数据发往服务器
-  正在从服务器下载返回数据
-  请求过程中暂停
-  返回过程中暂停
-  请求中使用了HTTP HEAD方法，返回中应该没有body内容
-  请求中使用了HTTP CONNECT方法，建立HTTPS连接通道
-  返回的内容类型是HTML
-  返回的内容类型是图片
-  返回的内容类型是Javascript
-  返回的内容类型是CSS
-  返回的内容类型是XML
-  普通的成功的返回
-  返回内容为 HTTP/300,301,302,303 or 307 跳转
-  返回内容为HTTP/304: 使用本地缓存
-  返回内容为一个证书请求
-  返回内容是服务器错误
-  请求被客户端、Fiddler或服务器中断

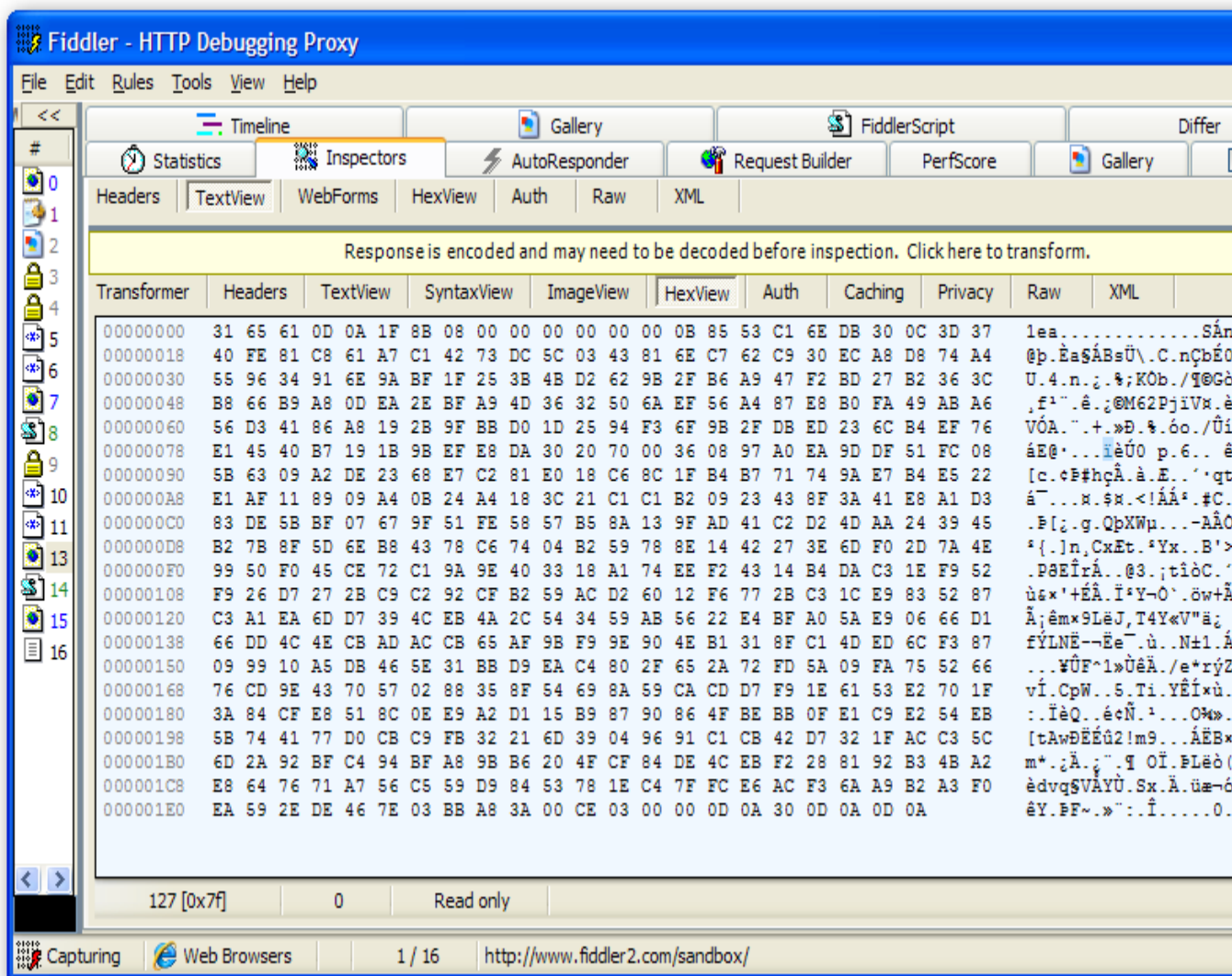
Fiddler 请求相关信息对应的主要功能：

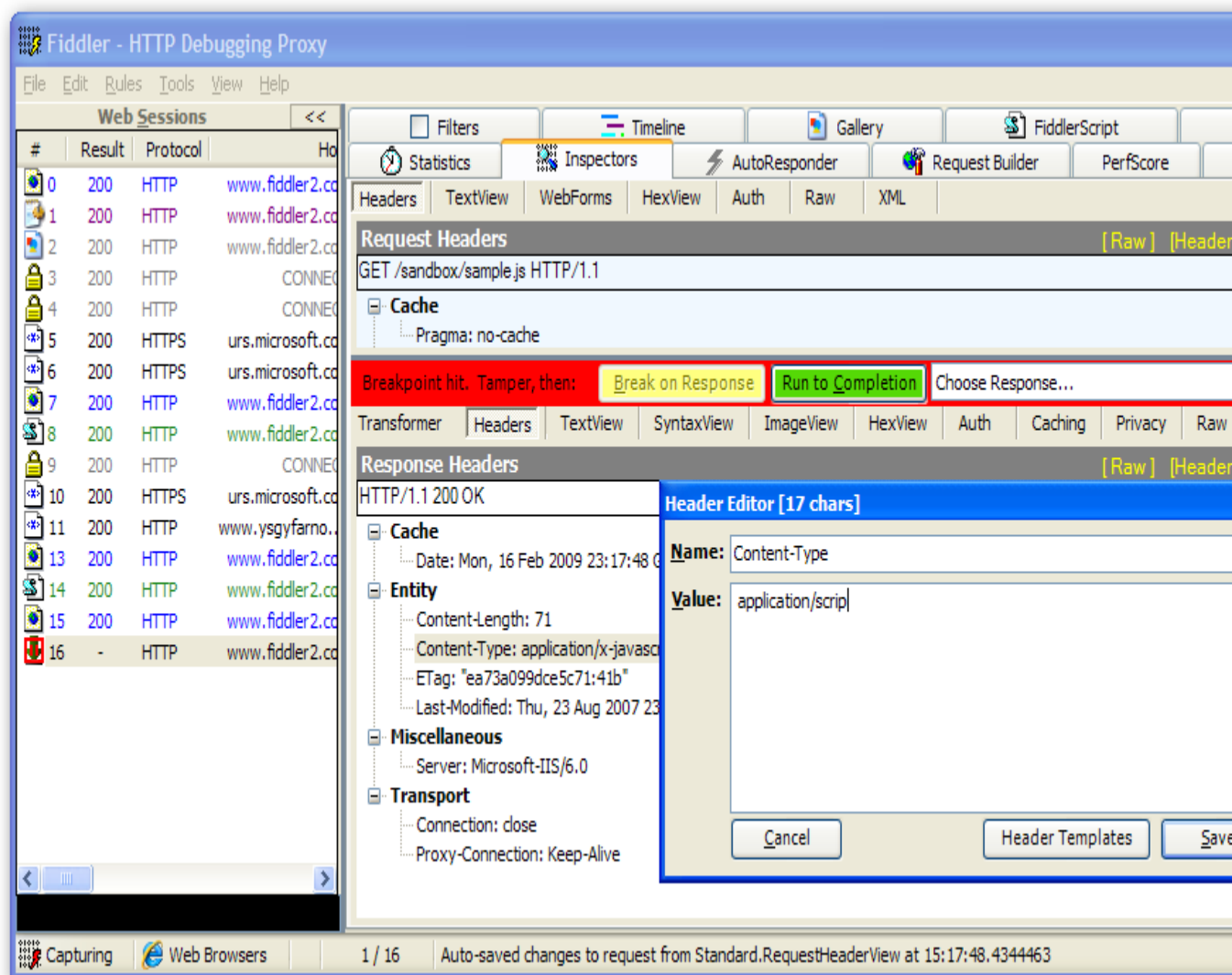
工具最右方的是请求相关信息的查看器，提供了数据多方面的查看方式。想了解？看图片。

统计资料信息(Statistics)

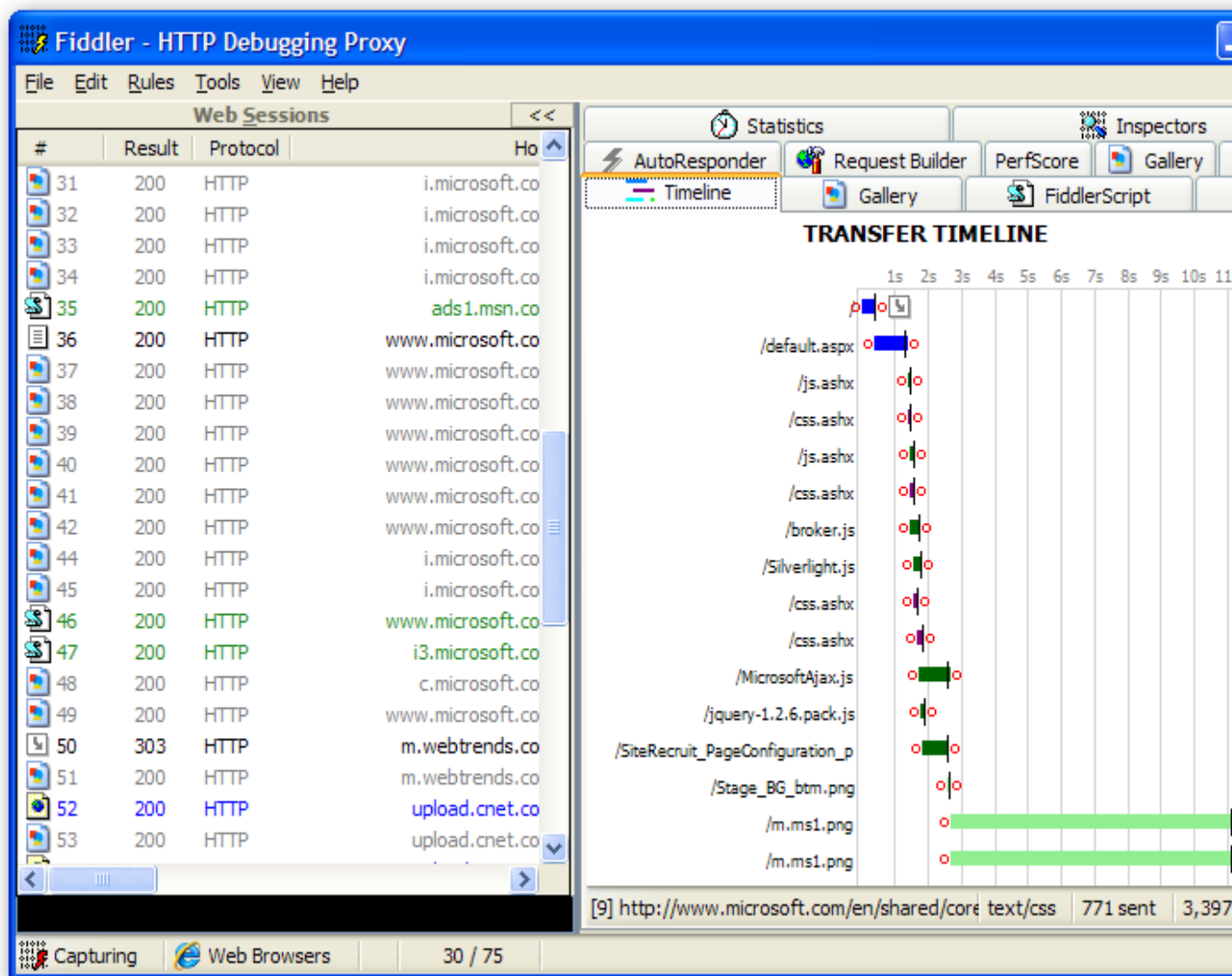


强大的检查器(Inspectors) - 功能很多, 等待你慢慢挖掘。

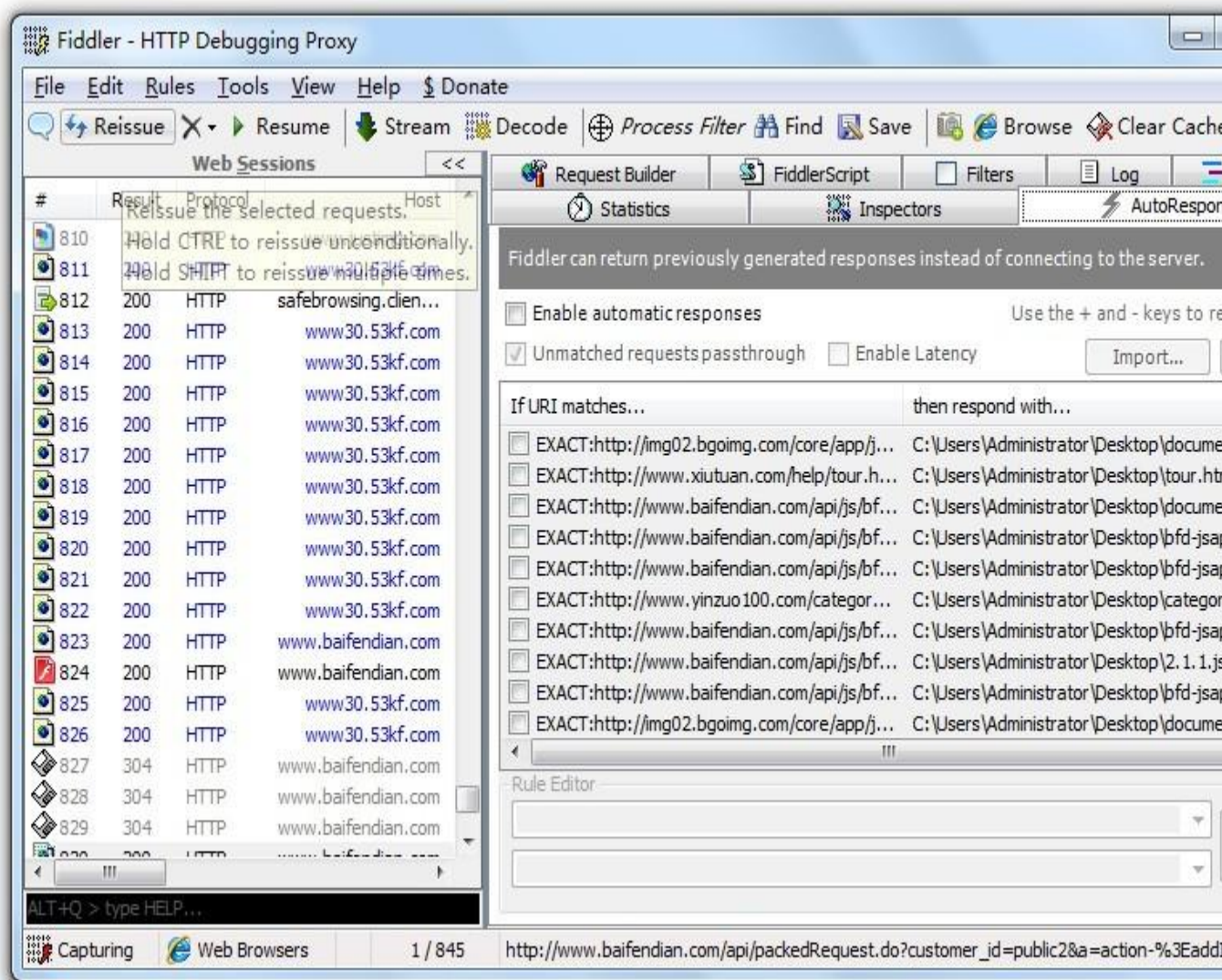




时间轴(Timeline)



自动回复器(autoResponder) - 一会就是介绍它的具体使用方法



说说我在工作中为什么使用 Fiddler，如何使用 Fiddler。

前端工程师在工作中总会有那么一些要求，要求书写的代码具有优良的兼容性，要求考虑代码的高性能，要求方法要面向对象，要求... 前端工程师总是和浏览器兼容有很多不得不说的。

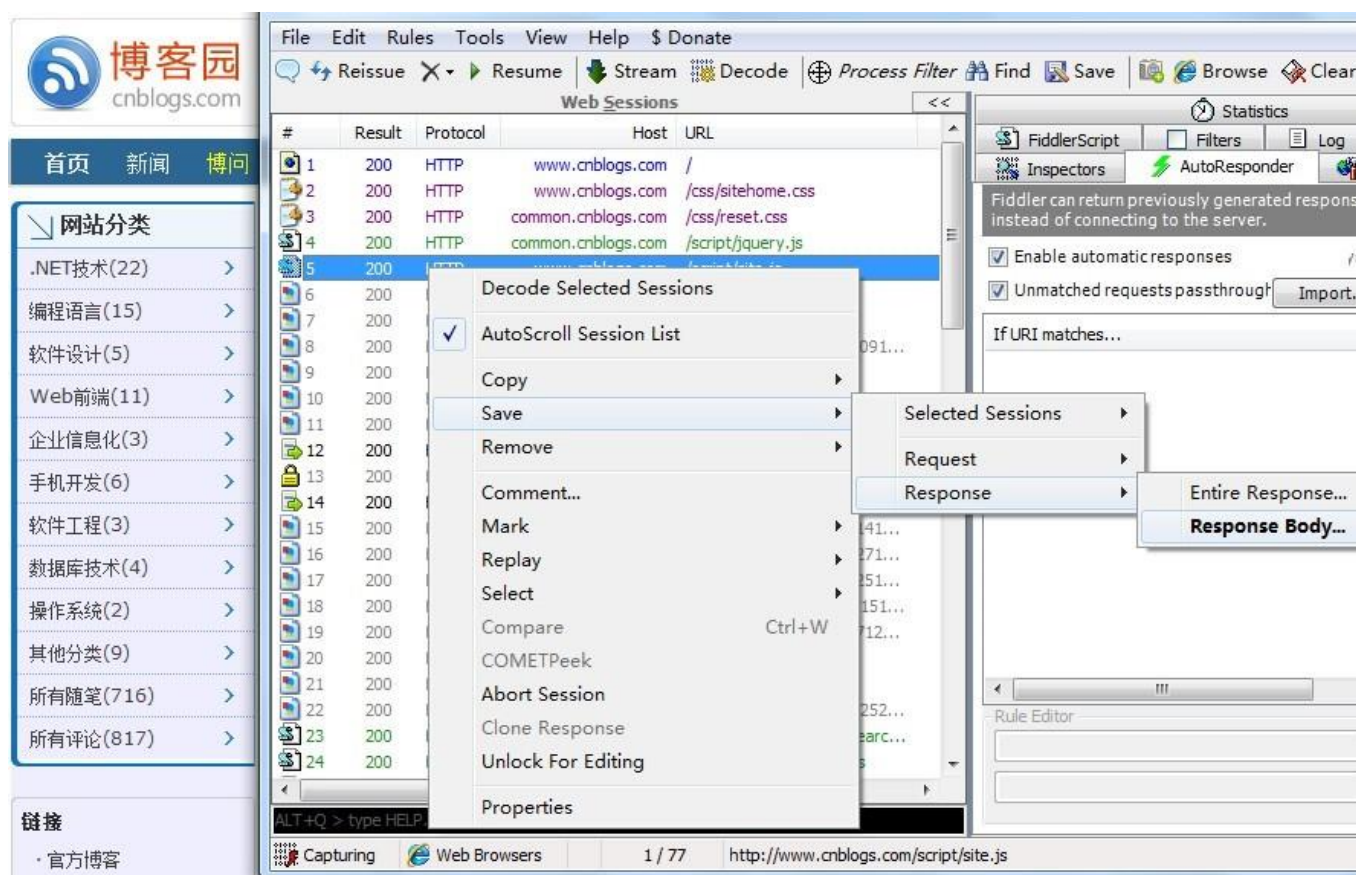
条件 1：在我们前端工程师开发的工作中，要调试服务器上某个 HTML/CSS/JavaScript 文件。一般情况下，我们都是将文件直接进行修改，然后重新发布再去做验证，这样就容易影响到测试环境或者生成环境的稳定性。更好的做法是，我们在本地开发环境中直接修改文件并进行验证，然后发布到测试环境，这样能保证测试环境的稳定，可是又比较繁琐。

条件 2：现在我的情况是需要调试上线产品的浏览器兼容性问题，且我没有本地环境或者生成环境去测试。假如有 Bug 发生在 Firefox 或者 Chrome 这种有控制台支持调试的浏览器下一切都好说，可是假如 bug 只发生在遨游，TT，世界

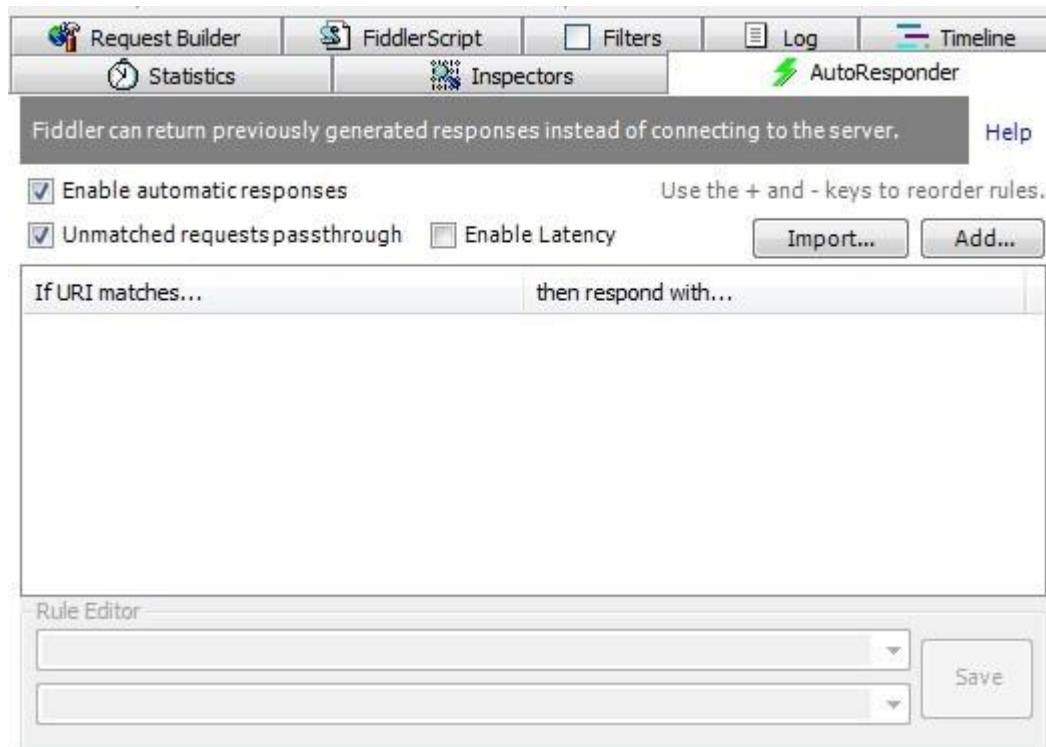
之窗, 搜狗... 这种的没有调试功能的浏览器下, 而且你 还碰见了我目前的情况, 那么如果没有 Fiddler 这种工具, 只能说这就是一场灾难。

Fiddler 工具可以修改 HTTP 数据的特性, 我们就非常便捷地基于生产环境修改并验证, 确认后再发布。

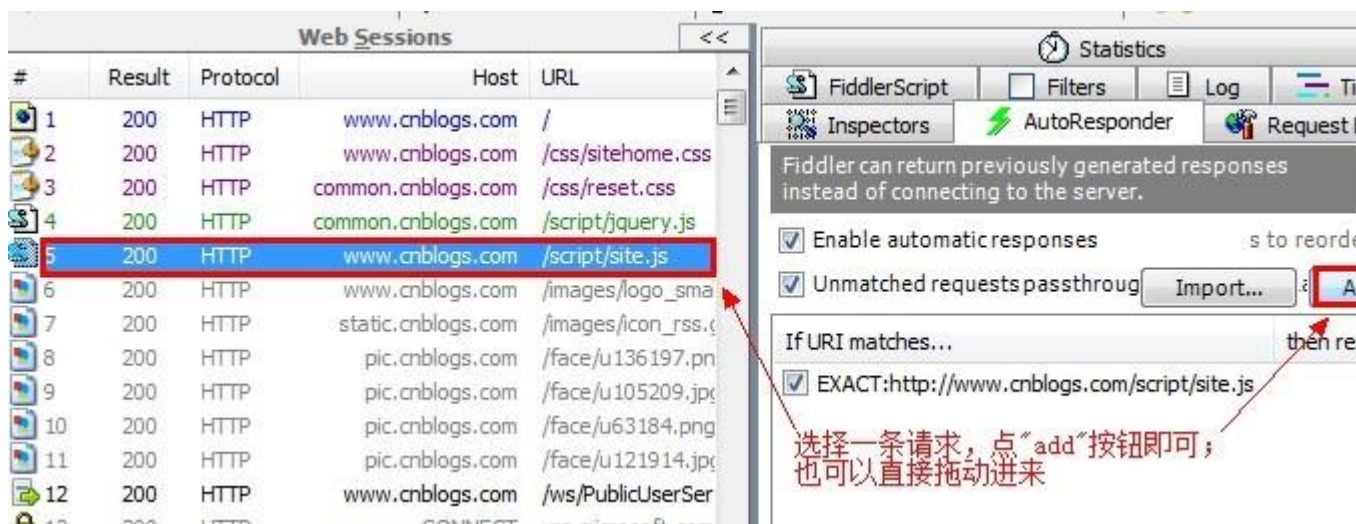
第一步, 先定位调试文件且下载。假设发现页面中的某个文件有问题 (HTML/CSS/JavaScript 都行), 那么我们需要做的是就把他先下载到本地 (如果本地有这个本地那么可以跳过此步骤), 下载到本地的文件偶尔会有乱码的情况, 建议你先清理浏览器缓存或者调整注册表 ([Fiddler2 中文乱码问题](#))。使用细节如下:



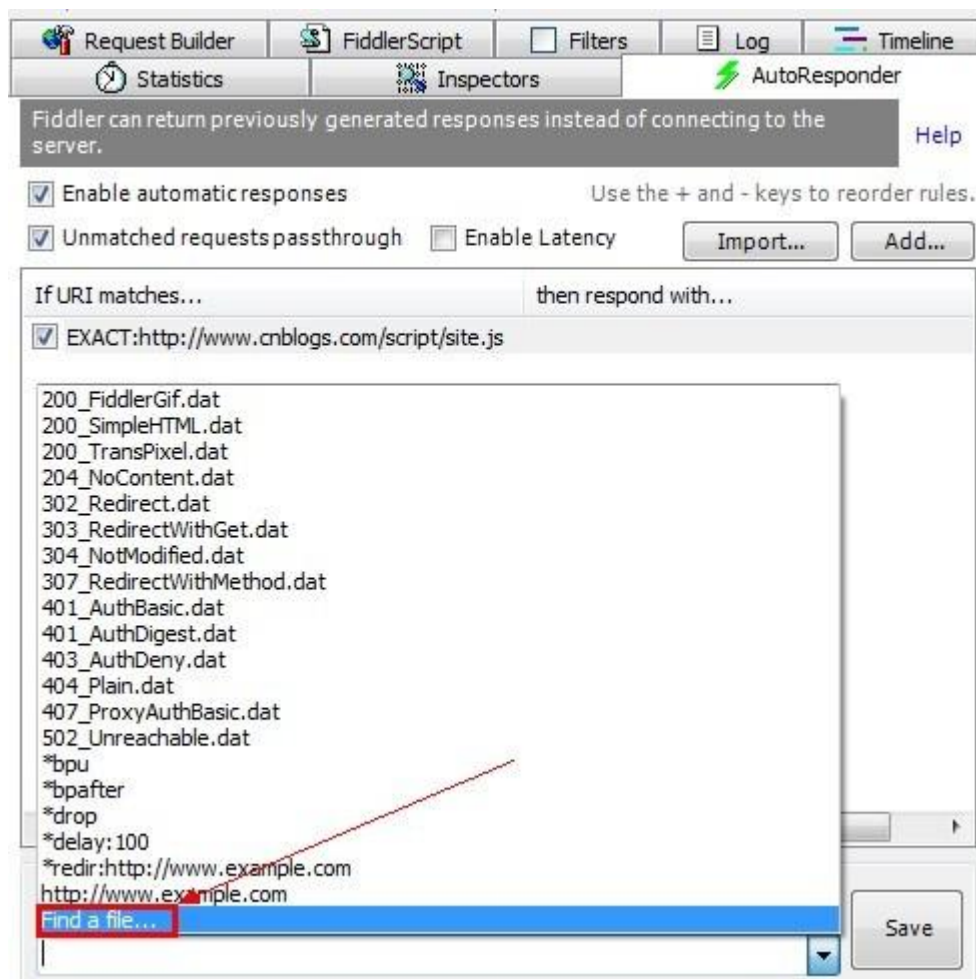
第二步, Fiddler - autoResponder 出场, 开启此功能。打开 AutoResponder 标签设置。可以看到界面上有三个选择框, 第一个的作用是开启或禁用自动重定向功能, 我们就可以在下面添加重定向规则了; 第二个选择框被勾上时, 不匹配的请求可以通过, 不影响那些没满足我们处理条件的请求。



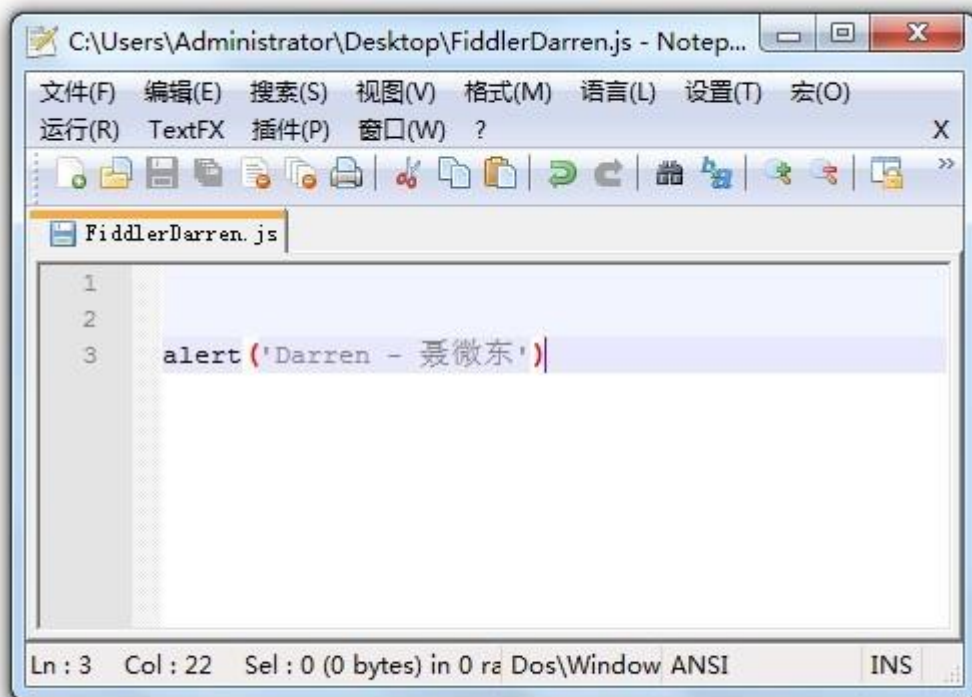
第三步，创建重定向规则，将目标是这个 js 的 HTTP 请求重定向到本地文件。选中刚刚定位的文件，通过“Add...”按钮增加规则，也可以直接拖动过来。



第四步，选择本地刚刚保存的文件或者替换的文件，作为替换这个请求的内容。



第五步，你调试或者不调试，它就在那里 - 只会请求你本地的选择的那个文件。所以，想怎么修改都随便你了。刷新页面，就可以看见这个 alert 了。



总结：虽然介绍时一共分为 5 个步骤，其实只要用习惯了很随意就可以调试了。[快速前端调试其实很简单](#)，你说类。

