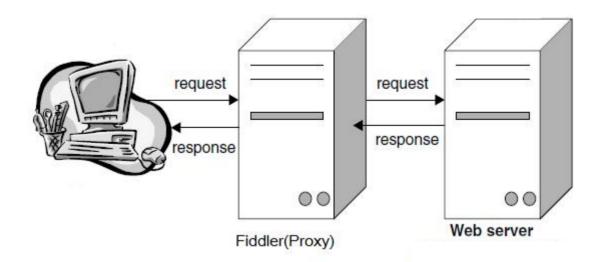
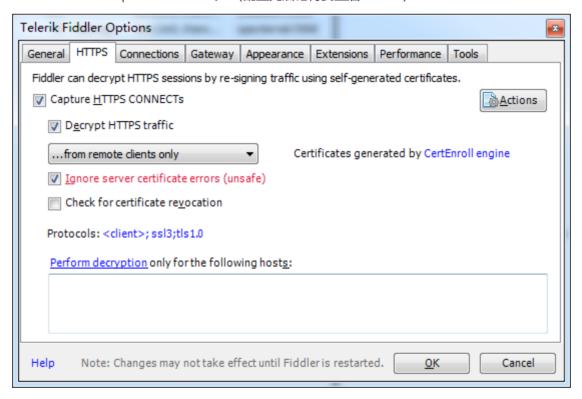
fiddler抓包原理



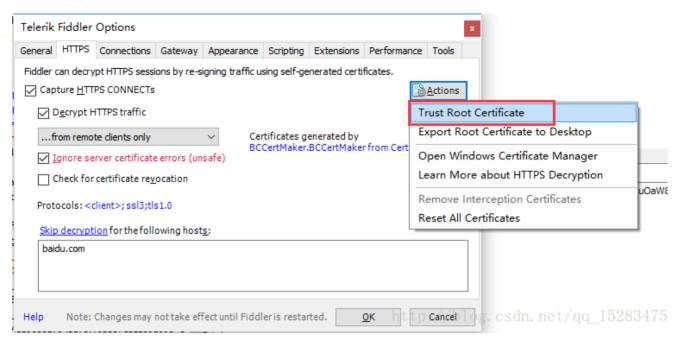
注意: Fiddler 是以代理web服务器的形式工作的,它使用代理地址:127.0.0.1,端口:8888。当Fiddler退出的时候它会自动注销,这样就不会影响别的程序。不过如果Fiddler非正常退出,这时候因为Fiddler没有自动注销,会造成网页无法访问。解决的办法是重新启动下Fiddler。

配置

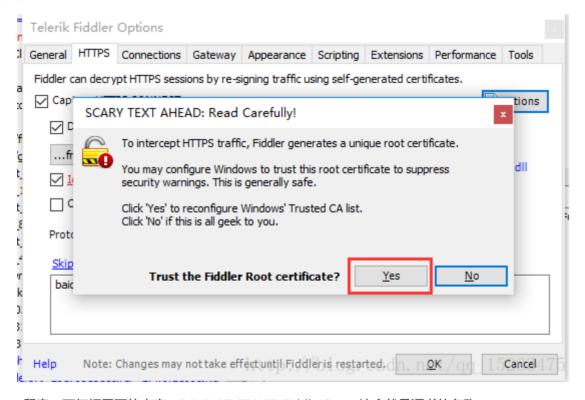
打开Fiddler Tool->Fiddler Options->HTTPS。 (配置完后记得要重启Fiddler).



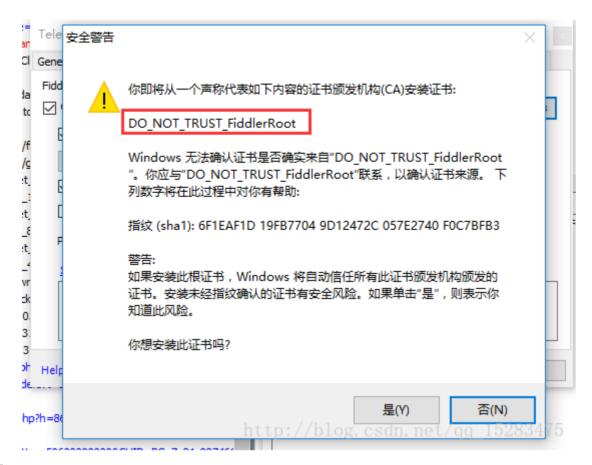
选中"Decrpt HTTPS traffic", Fiddler就可以截获HTTPS请求,第一次会弹出证书安装提示,若没有弹出提示,勾选 Actions-> Trust Root Certificate 另外,如果你要监听的程序访问的 HTTPS 站点使用的是不可信的证书,则请接着把下面的 "Ignore servercertificate errors" 勾选上。



证书安装提示:



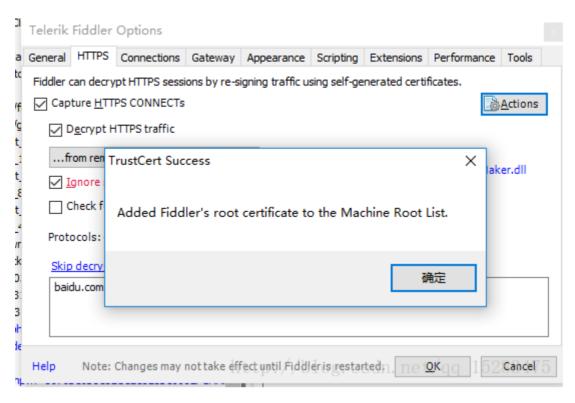
点击Yes,留意一下红框里面的内容,DO_NOT_TRUST_FiddlerRoot,这个就是证书的名称。



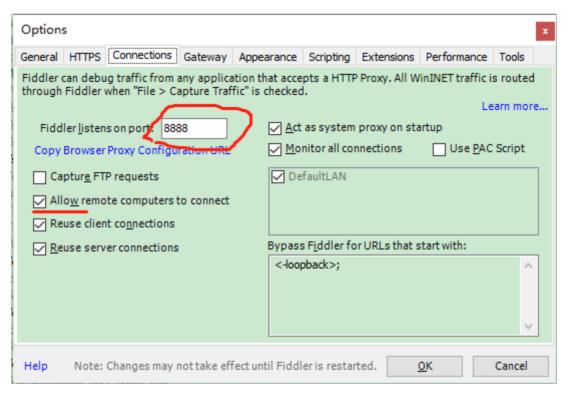
点击是。



点击确定,这样Fiddler证书就已经添加成功了。



手机端抓包配置



fiddler监听端口默认是 8888,你可以把它设置成任何你想要的端口。勾选上 "Allow remote computersto connect",允许远程设备连接。

为了减少干扰,可以去掉 "Act assystem proxy on startup"。

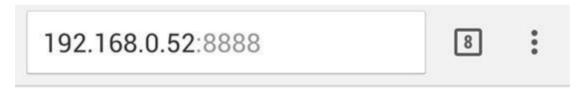
手机端(客户端)设置

首先查看电脑的 IP 地址 (ipconfig) ,确保手机和电脑在同一个局域网内

```
连接特定的 DNS 后缀 . . . . . . :
本地链接 IPv6 地址. . . . . . : fe80::f ;:f13e:1b0c:bae7%12
IPv4 地址 . . . . . . . . : 192.168 . . 1
子网掩码 . . . . . . . . . : 255.25' 55.0
默认网关. . . . . . . . . . :
```

Android 手机上的配置

将 Fiddler 代理服务器的证书导到手机上才能抓这些 APP 的包。导入的过程:打开浏览器,在地址栏中输入代理服务器的 IP 和端口(即电脑的IP加fiddler的端口),会看到一个Fiddler 提供的页面,然后确定安装就好了



Fiddler Echo Service

GET / HTTP/1.1
Host: 192.168.0.52:8888
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (Linux; Android 4.2.1; Coolpad8720Q Build/JOP40D) AppleWebKit/537.36 (KHTML, like Gecko) Ch Accept-Encoding: gzip, deflate, sdch
Accept-Language: zh-CN,zh;q=0.8

This page returned a HTTP/200 response

- . To configure Fiddler as a reverse proxy instead of seeing this page, see Reverse Proxy Setup
- · You can download the FiddlerRoot certificate

点击下载证书

打开 WiFi 设置页面,选择要连接的 wifi ,并且长按,在弹出的对话框中,选择"修改网络"。在接下来弹出的对话框中,勾选"显示高级选项"。在接下来显示的页面中,点击"代理",选择"手动"。代理服务器主机名设为 PC 的 IP ,代理服务器端口设为 Fiddler 上配置的端口 8888,点"保存"。



苹果手机上的配置

苹果手机上的配置其实跟 Android 手机基本是一样的。如图





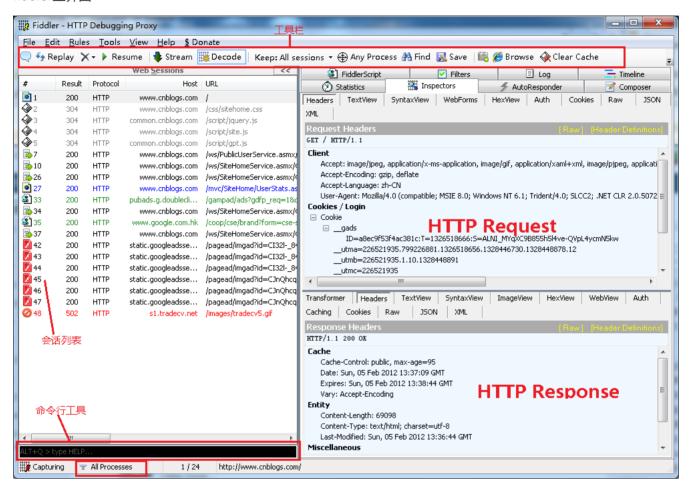
至此已配置完成。

Fiddler的使用

视图功能区域

会话的概念:一次请求和一次响应就是一个会话。

fiddler主界面



下面挑几个快捷功能区中常用几项解释,其他功能自己尝试:



快捷功能区

- 1: 给会话添加备注信息
- 2: 重新加载当前会话
- 3: 删除会话选项
- 4: 放行
- 5:响应模式。也即是,当Fiddler拿到远程的response后是缓存起来一次响应给客户端还是以stream的方式直接响应。
- 6:解码。有些请求是被编码的,点击这个按钮后可以根据响应的编码格式自动解码。
- 7: 查找会话。
- 8: 保存会话。
- 9: 截屏。截屏后,会以会话的方式返回一个截图。

接着来看看会话列表

- 1. [#] -- HTTP Request 的顺序,从 1 开始,按照页面加载请求的顺序递增。
- 2. **[Result]** —— HTTP 响应的状态,可以参考这里。
- 3. [Protocol] —— 请求使用的协议(如 HTTP/HTTPS/FTP)
- 4. [Host] —— 请求地址的域名
- 5. [URL] —— 请求的服务器路径和文件名,也包括 GET 参数
- 6. **[BODY]** —— 请求的大小,以 byte 为单位
- 7. [Caching] —— 请求的缓存过期时间或缓存控制 header 等值
- 8. **[Content-Type]** —— 请求响应的类型(Content-Type)
- 9. [Process] —— 发出此请求的 Windows 进程及进程 ID
- 10. [Comments] —— 用户通过脚本或者右键菜单给此 session 增加的备注
- 11. [Custom] —— 用户可以通过脚本设置的自定义值

#栏图标说明

- ↑ -- 请求已被发送到服务器
- ➡ -- 从服务器下载响应结果
- 🛅 —— 请求在断点处被暂停
- 🗓 —— 响应在断点处被暂停
- 一 请求使用 HTTP HEAD 方法,响应没有内容
- 🔒 —— 请求使用 HTTP CONNECT 方法,使用 HTTPS 协议建立连接通道
- 遂 -- 响应是 HTML 格式
- 🛐 -- 响应是图片格式
- -- 响应是脚本文件/blog.csdn.net/ohmygirl
- 🥦 -- 响应是 CSS 文件
- 一一 普通响应成功
- Ы -- 响应是 HTTP 300/301/302/303/307 转向
- → 一 响应是 HTTP 304 (无变更),使用缓存文件
- 🪏 —— 响应需要客户端验证
- ▲ -- 响应是服务器错误
- 一 请求被客户端、Fiddler或者服务器终止(Aborted)