APP专项性能测试

我们以全民K歌为例

# 启动时间

## 什么是应用启动时间

* 从点击应用的启动图标开始创建出一个新的进程直到我们看到了界面的第一帧，这段时间就是应用的启动时间
* 启动时间是用毫秒度量
* 三种启动方式
* app冷启动：当应用启动时，后台没有该应用的进程，这时系统会重新创建一个新的进程分配给该应用，这个启动方式就叫做冷启动（后台不存在该应用进程）。
* app热启动：当应用已经被打开，但是被按下返回键、Home键等按键时回到桌面或者是其他程序的时候，再重新打开该app时，这个方式叫做热启动（后台已经存在该应用进程）。、
* app首次启动:首次启动应用时会有很多的数据或配置文件的初始化工作，因此其启动时间远远大于冷启动的启动时间，这样的话app的启动时间也就变成了从3个指标去衡量了

## 具体操作

### 命令，初始启动

|  |
| --- |
| adb shell am start -W [packageName]/[packageName.MainActivity] |

-W：wait for launch to complate

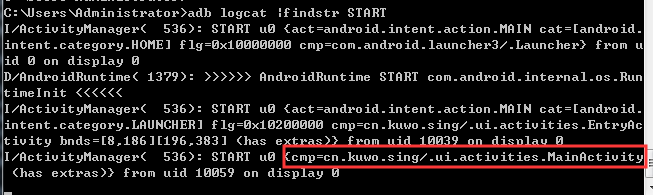


### 先获取包名，在初始启动

输入如下命令后，点击具体的应用

|  |
| --- |
| adb logcat | findstr START |

成功后看到具体时间参数

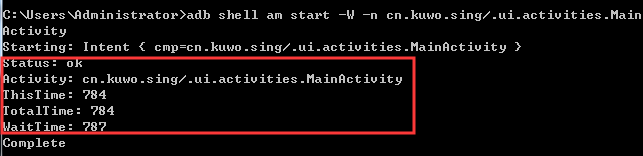


### 强行关闭程序

|  |
| --- |
| adb shell am force-stop cn.kuwo.sing |

### 执行命令，获取启动时间

|  |
| --- |
| adb shell am start -W -n cn.kuwo.sing/.ui.activities.MainActivity |



## 分析

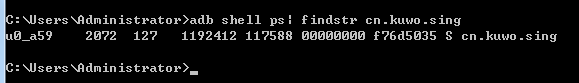
执行成功后将返回三个测量到的时间：

* ThisTime：一般和TotalTime时间一样，除非在应用启动时开了一个透明的Activity预先处理一些事再显示出主Activity，这样将比TotalTime小。
* TotalTime：应用的启动时间，包括创建进程+Application初始化+Activity初始化到界面显示。
* WaitTime：一般比TotalTime大点，包括系统影响的耗时。

# 流量监控

## 获取进程id

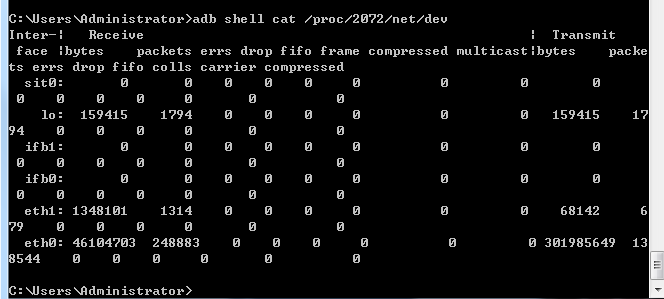
|  |
| --- |
| adb shell ps | findstr 包名 |



在这里，2072（第二列）为具体进程号（和Linux查pid一样）

## 获取该进程流量

|  |
| --- |
| adb shell cat /proc/pid/net/dev |



* Recevice:表示收包
* Transmit:表示发包
* bytes: 表示收发的字节数
* packets: 表示收发正确的包量
* errs: 表示收发错误的包量
* drop: 表示收发丢弃的包量

## 注意

1、如果想得到某个业务所消耗的流量,就可以操作业务步骤后再次获取流量数据

2、如果想要将数据清0,开关一下飞行模式,再执行命令即可

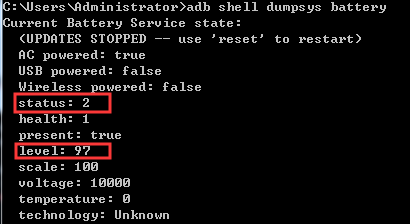
## 计算流量消耗

(将上下行的数据相加-测试前的数据)/1024/1024就得到所消耗多少M的流量

# 电量测试

## 命令

|  |
| --- |
| adb shell dumpsys battery |



需注意，手机通过USB连接电脑，手机会处于充电状态，必须保证手机为非充电状态

## 切换非充电状态

|  |
| --- |
| adb shell dumpsys battery set status 1 |

status 1 为非充电状态，status 2 为充电状态

## 切换为充电状态

|  |
| --- |
| adb shell dumpsys battery set status 2 |

# adb shell dumpsys详解

## battery：电池状态

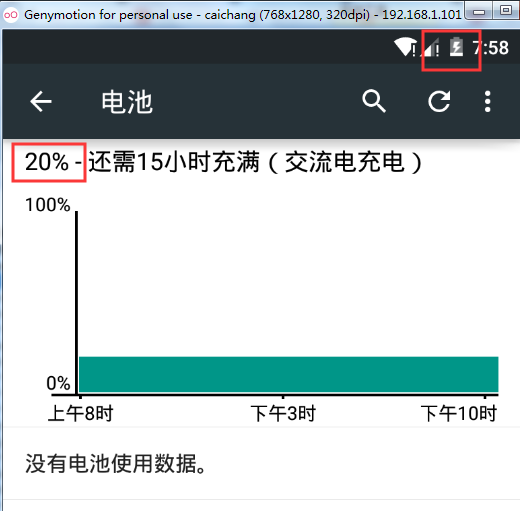
|  |
| --- |
| C:\Users\Administrator>adb shell dumpsys battery level 20  Dump current battery state, or:  set [ac|usb|wireless|status|level|invalid] <value>  reset |

我们可以看到，使用的是后 set 参数 值 的方式

于是敲，可以看到马上就发生了变化

|  |
| --- |
| C:\Users\Administrator>adb shell dumpsys battery set level 20 |





## batteryproperties：电池属性

## batterystats：电池状态

## cpuinfo：cpu信息

## devicestoragemonitor：设备存储监视器

## diskstats：磁盘统计

## ioinfo：I/O信息

## meminfo：内存信息

## mount：挂载信息

## netstats：网络统计

## network\_management：网络管理

## package：包信息

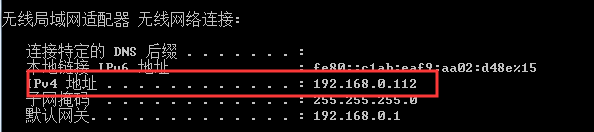
## processinfo：进程信息

## procstats：进程统计

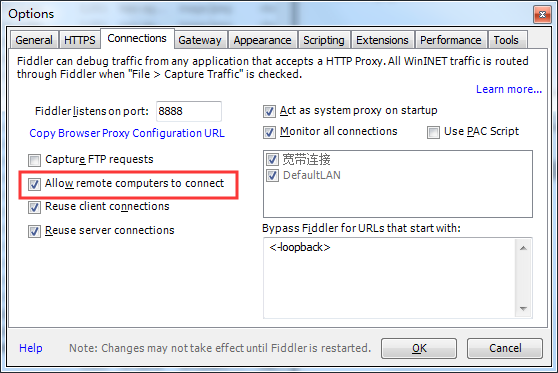
## gfxinfo：有关与录制阶段期间发生的动画帧相关的性能信息

# fiddler抓app包

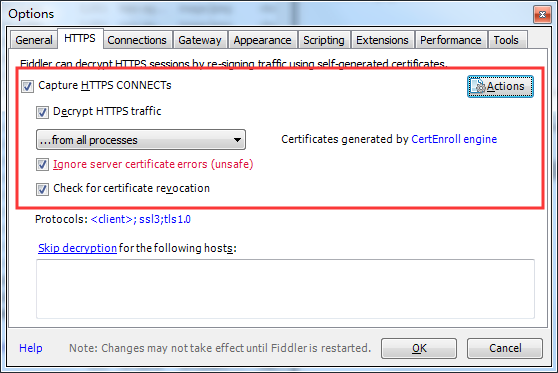
1. 查看本级ip，主要是哪个网卡上网就查哪台，千万别弄错了。我这里是通过无线上网，所以查无线网卡的ip



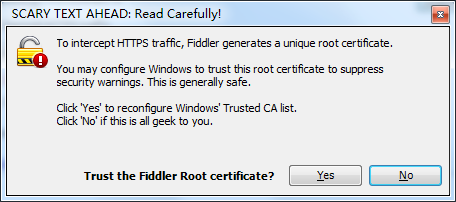
1. fiddler设置
   1. tool--options--connections--勾选allow remote computers toconnect（允许远程连接）



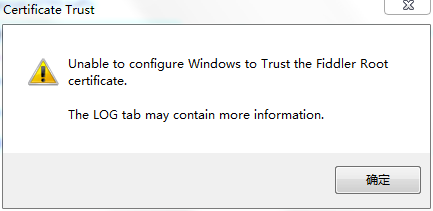
* 1. 勾选如下选项



点击actions，选择第一项，会弹出警告，选择yes

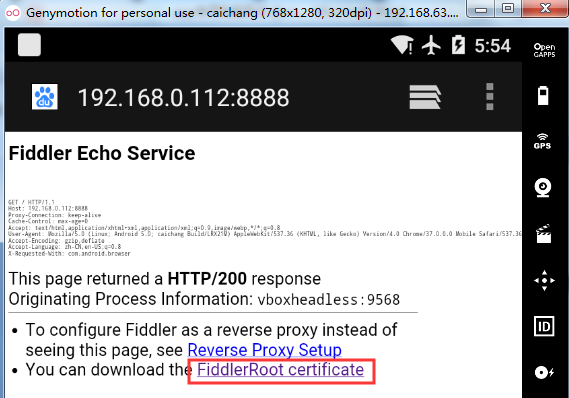


接着点击确定

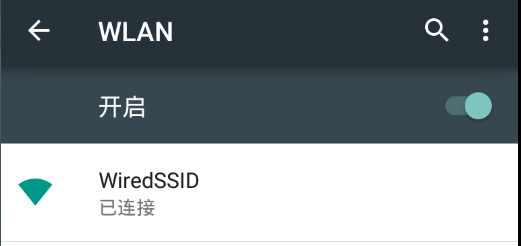


* 1. **重启fiddler（这点非常重要，很多人抓不到包就是没有先重启fiddler）**

1. 虚拟机or手机设置
2. 访问http://192.168.0.112:8888,即windows上ip+fiddler设定的端口，安装证书



1. 在wifi设定wifi的代理。按住这个不边，选择修改网路，设定豪代理，注意ip和端口。ip为windows无线网卡ip、端口为fiddler远程端口





1. 打开浏览器输入www.baidu.com,如发现如下错误，打开dos进入fiddler目录执行如下操作，当出现succeeded就不会再弹错误了

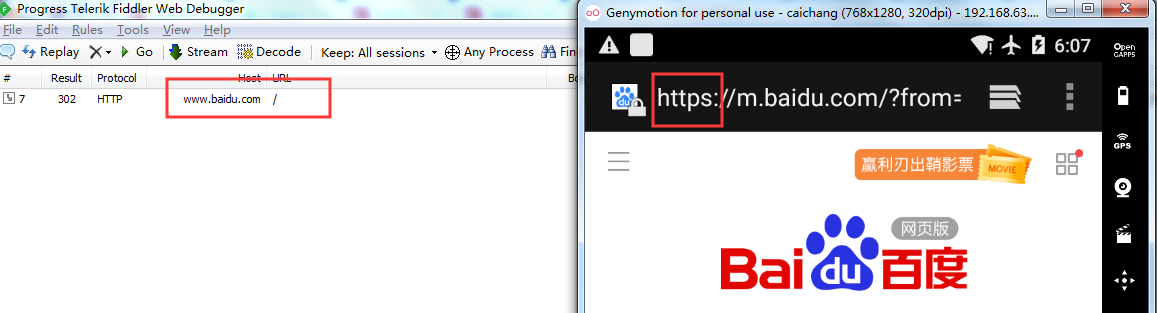
|  |
| --- |
| makecert.exe -r -ss my -n "CN=DO\_NOT\_TRUST\_FiddlerRoot, O=DO\_NOT\_TRUST, OU=Created by http://www.fiddler2.com" -sky signature -eku 1.3.6.1.5.5.7.3.1 -h 1 -cy authority -a sha1 -m 120 -b 09/05/2012 |



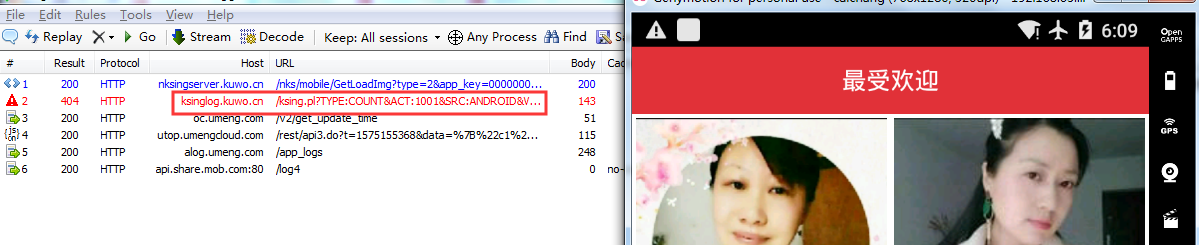
1. 再访问http://192.168.0.112:8888，就会弹出让你安装证书，确定后就会告知证书安装完成

1. 再执行百度，可以看到已抓到包



1. 执行安装的酷我K歌app，发现也能抓到包

不但如此，还发现一个404的bug，从代码看，是perl写的，有点意思：）