

APP测试

一、APP平台

安卓：底层是C/C+，应用层使用java语言编写，安装包是 **.apk**结尾

IOS：开发语言swift，安装包名 **.ipa**

鸿蒙：目前上层框架还是以安卓开源系统为主(底层全新开发系统)，兼容了市场上所有的安卓应用，安装包**.apk**

二、测试对象(主体)

1、应用：

安卓应用、iOS应用

2、H5页面

简单理解为网页，支持跨平台使用（手机、平板、PC浏览器）

H5页面的体现：

公众号：**微信公众号**、支付宝公众号、百度公众号

小程序：**微信小程序**、支付宝小程序、百度小程序

3、快应用

手机厂商开发的小程序

无需安装，即点即用，具备传统APP完整的应用体验

(<https://cdn.cnbj1.fds.api.mi-img.com/quickappcenter1/>小米快应用介绍.pdf)



4、手机网站

m.jd.com

xw.qq.com

三、APP项目的测试流程

跟其他web项目的测试流程是一样的。

1、需求大会/分析

同步需求，达成一致。参与人员：**开发**(含架构师)、**测试**、**产品**经理、项目经理、运维(知道什么上线发布)

2、测试计划

测试方面的计划：设备资源、测试范围、人员安排、时间安排

3、编写测试用例

按测试人员的分工进行编写

4、评审测试用例

内部：测试组内大概过一下

外部评审：项目组内，**测试组织**，**开发**、**产品**、项目、客户

5、执行测试用例

按测试用例执行---出现BUG---提交BUG--开发修改BUG--再回归验证BUG是否解决

6、编写测试报告

测试报告内容：测试环境说明，测试的范围、测试时间、执行的用例数、BUG数以及分布情况，是否通过、建议等

7、APP项目上线发布

上线时间，挑用户量少的时候。

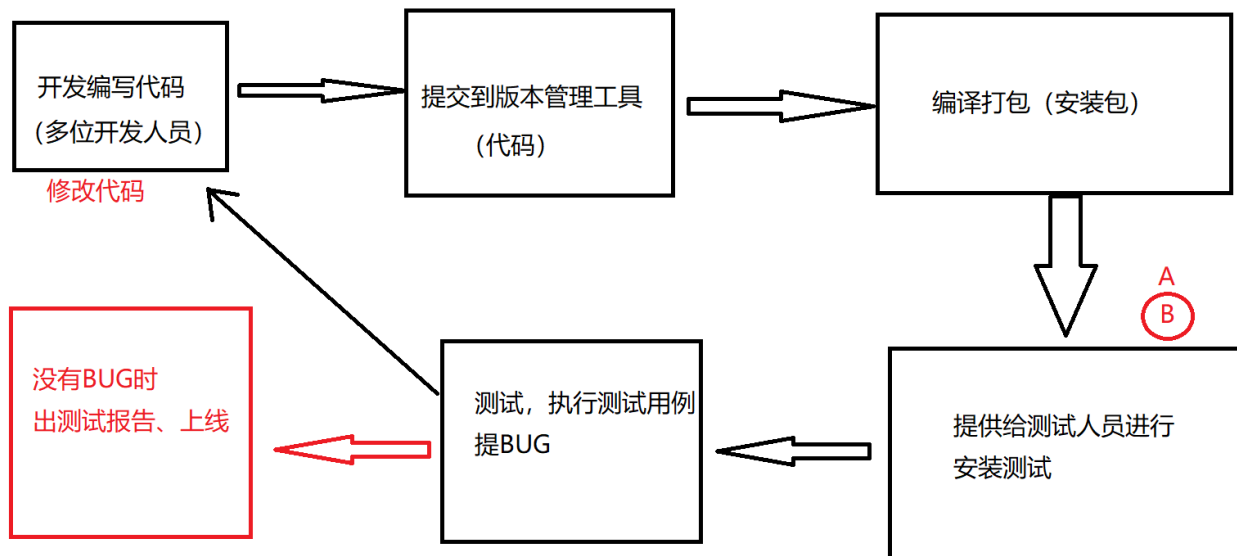
上线发布过程中出现BUG怎么办？按BUG等级、修复难易程度来说

修复难度高，影响程度深---沟通是否可以暂缓发布，等待修复后再上线

修复难度高，影响程度浅---跟项目经理、产品沟通是否可以先上线，下个版本在修复

修复难度低，影响程度深、浅---花费时间立即修复，修复后再上线

四、APP项目过程中，安装包的生产过程



备注：如果是APP客户端方面的BUG，开发解决bug后，需要重新安装新的APP包，才能进行回归验证BUG是否解决。

五、APP测试方法/种类

1、APP还是以功能测试为主（85%以上）

设计测试用例方法：等价类、边界值、因果图、流程分析法、状态迁移、正交表、错误推断....

（根据实际情况选择合适的设计方法）

2、安装、卸载测试

全新安装：手机上没有这个APP应用，全新安装，安装成功，APP可以正常使用

覆盖安装：已经存在了对应的APP应用，需要升级。

卸载：卸载APP应用后，APP应用数据是否删除干净

（应用卸载后，没有删除数据，给予其他人使用的时候，直接安装了是否就可以免密登陆？

拍照的应用，卸载时没有删除照片，那么其他人时候可以通过再次安装该应用获取到原来的用户数据？）

3、接口测试

接口是由后端开发提供出来的，给到客户端使用。

①、架构：C/S架构

web端：B/S

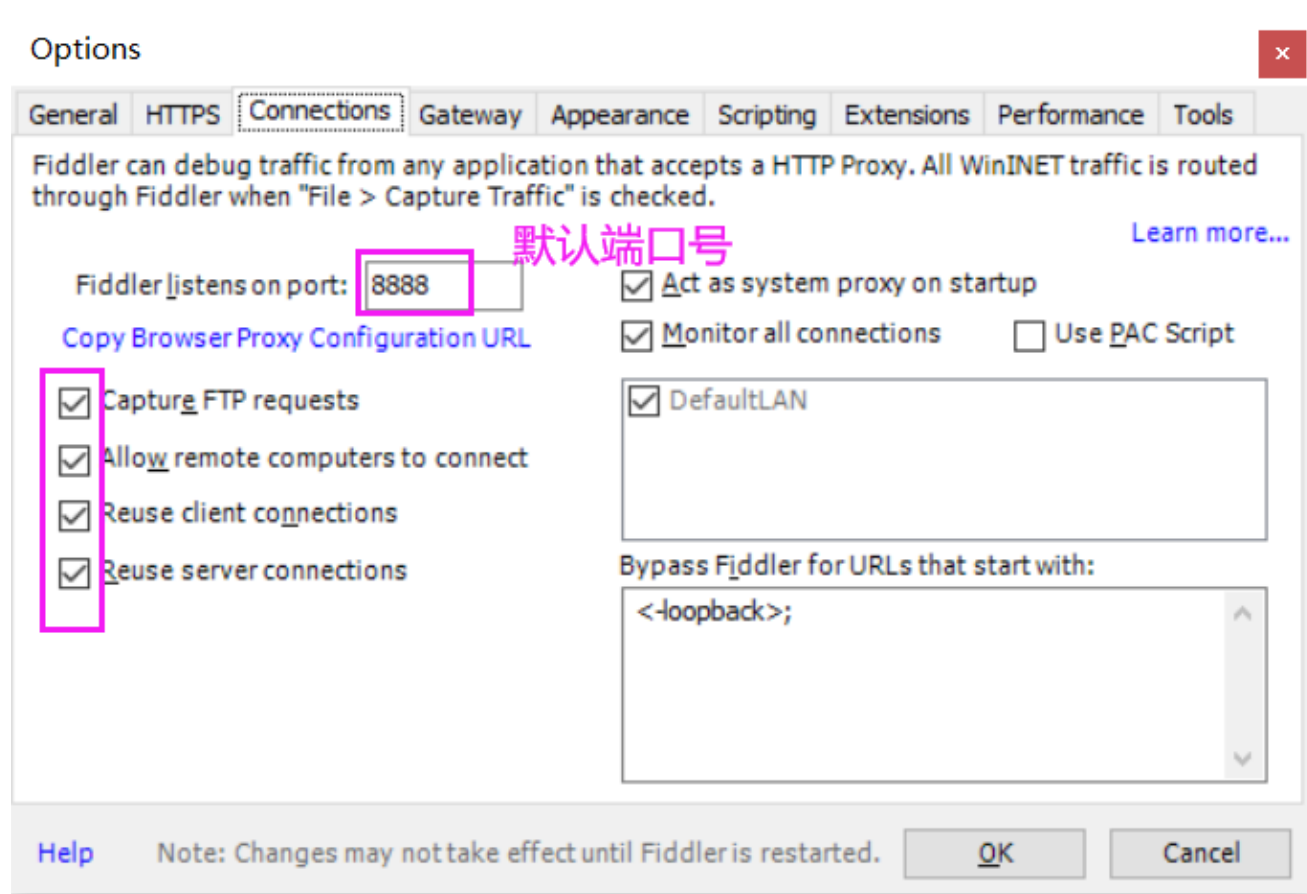
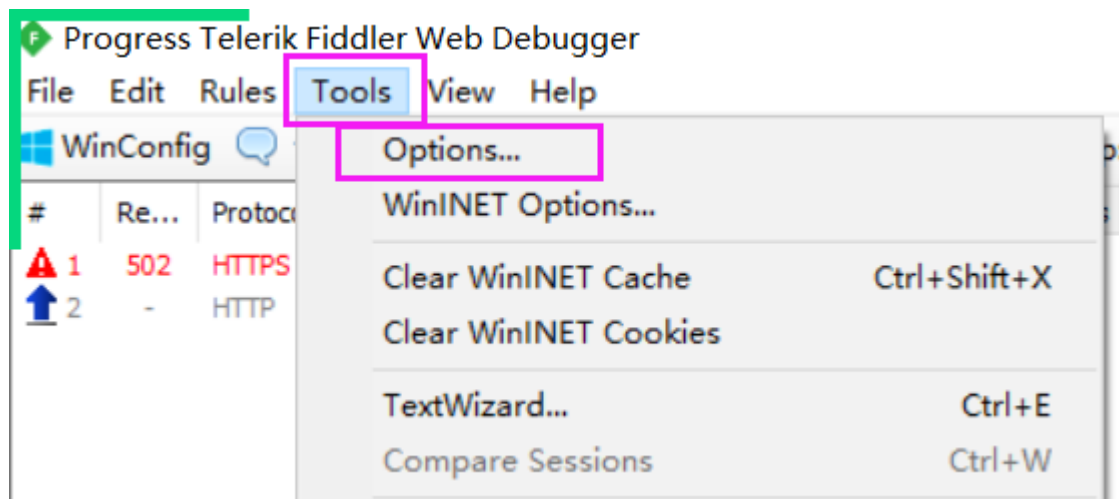
②、项目根据需求，进行接口测试

接口文档出来，在前后端开发联调阶段，在系统集成测试之前做接口测试。

③、APP抓包--fiddler

第一步：fiddler需要开启远程代理，设置默认端口号8888，重启fiddler生效

(代理服务：在本地电脑记录并转发请求跟响应)



第二步：移动设备需要跟电脑在同一局域网内

(如果是真实设备--真机，那么连接同一个路由WiFi即可)

如果是使用电脑上安装的模拟器，可以忽略这步)

第三步：查询本地电脑的IP地址

dos命令下：ipconfig

(查找默认网关存在内容，对应的IPV4地址)

```
以太网适配器 以太网 3:

    连接特定的 DNS 后缀 . . . . . : 
    本地链接 IPv6 地址. . . . . : fe80::3042:5ae2:580d:4a4c%15
    IPv4 地址 . . . . . : 192.168.0.43
    子网掩码 . . . . . : 255.255.255.0
    默认网关. . . . . : 192.168.0.1
```

IP: 192.168.0.43

第四步：真机/模拟器 长按网络进行修改

设置---WLAN---长按---修改网络---高级选项：

代理：改成手动代理

代理服务器主机名：电脑的IP地址

代理服务器端口：fiddler的远程代理端口号



038s56V48Opd

高级选项

代理

手动

浏览器会使用 HTTP 代理，但其他应用可能不会使用。

代理服务器主机名

192.168.0.43

填写电脑的IP地址

代理服务器端口

8888

填写fiddler远程代理的端口号

对以下网址不使用代理

.

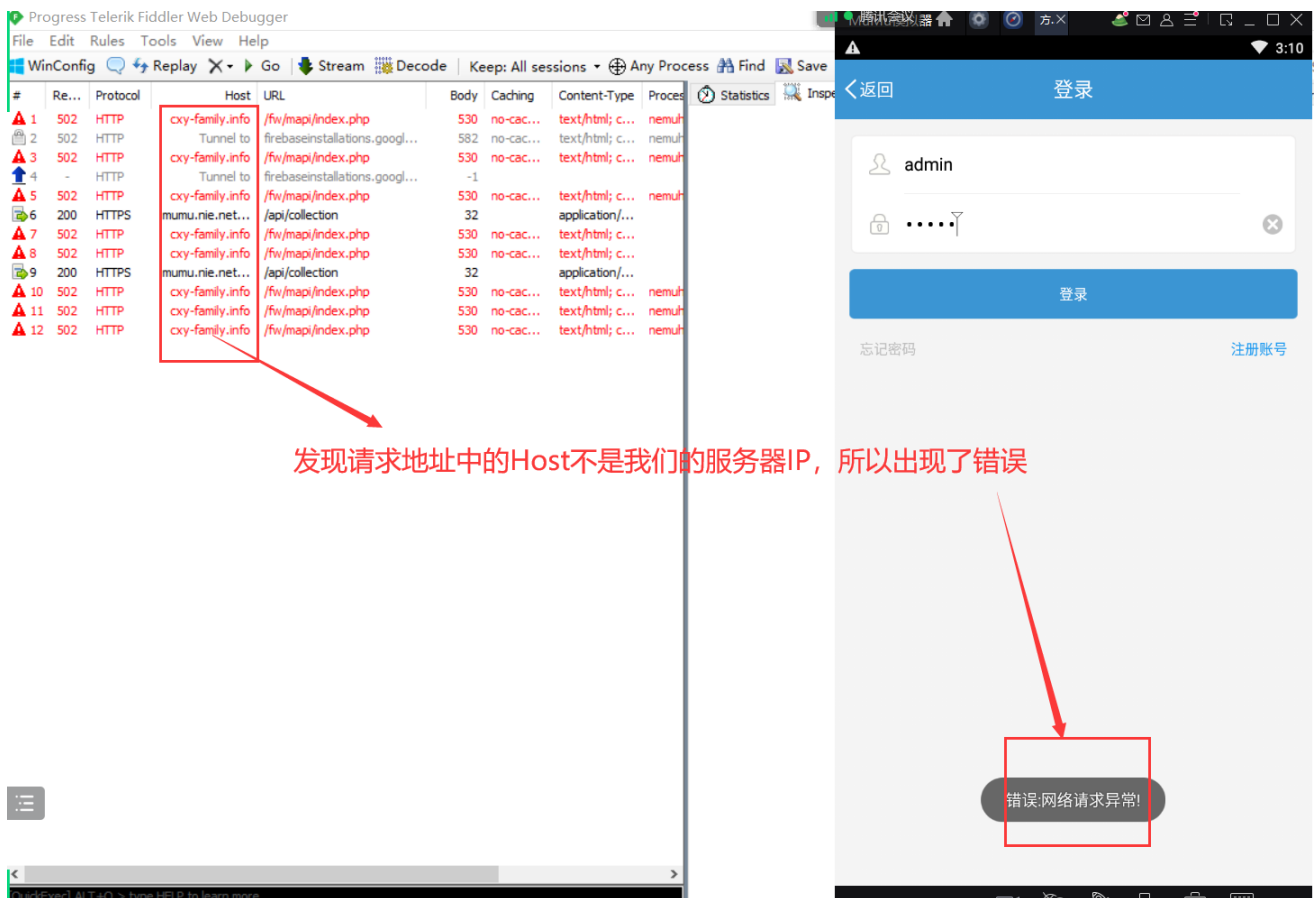
.

..

取消

保存

安装<方维P2P>APP包，操作后抓包查看，发现请求Host不是对应的服务器IP，导致出现的错误



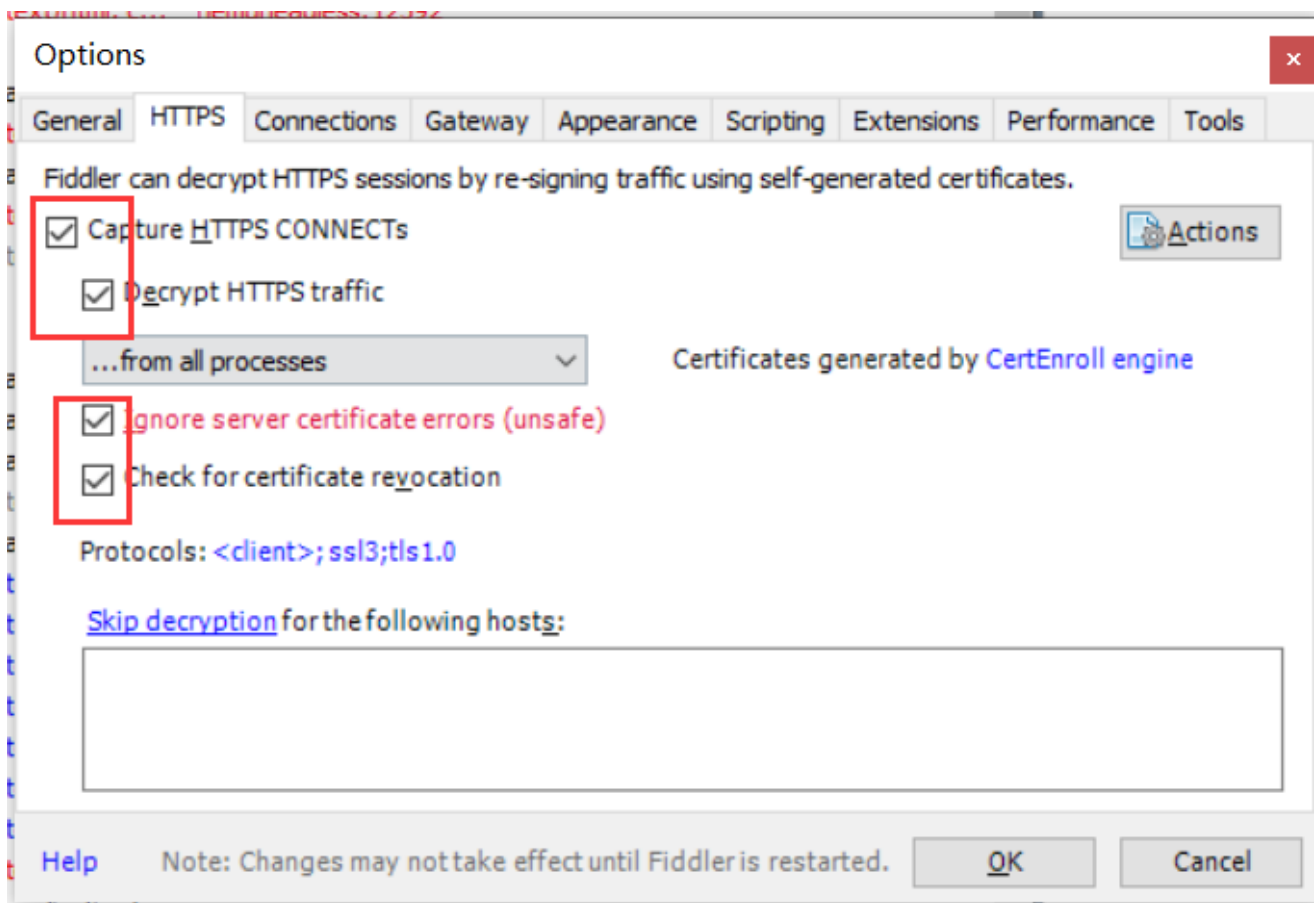
④、抓取APP中HTTPS的请求

能抓https，肯定能抓http请求，

想要抓取HTTPS，前面的步骤③的操作都是需要的

还需要增加两步操作：

额外步骤一、fiddler开启抓取https请求



额外步骤二：移动设备需要下载安装https证书：通过浏览器访问电脑IP地址+远程代理端口号进行下载安装



Fiddler Echo Service

```
3ET / HTTP/1.1
Host: 192.168.0.43:8888
Proxy-Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Linux; Android 6.0.1; MuMu Build/V417IR; wv) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/60.0.3112.101 Mobile Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,en-US;q=0.9
X-Requested-With: com.android.browser
```

This page returned a **HTTP/200** response
Originating Process Information: nemuheadless:12592

- To configure Fiddler as a reverse proxy instead of seeing this page, see [Reverse Proxy Setup](#)
- You can download the [FiddlerRoot certificate](#)

下载证书

下载完成，点击进行安装：安装过程中会提示设置密码(锁屏密码)，请务必牢记（统一设置：a123456）

为证书命名

证书名称：

自定义

凭据用途：

VPN和应用

该数据包包含：

1个CA证书

安装过程中会涉及到密码，请务必牢记

取消

确定

4、弱网测试

①、弱网在生活中的场景：

APP在实际使用过程中，时常会遇到网络不好的情况：城中村、地铁、隧道、电梯、火车站、基站切换

②、弱网测试的目的/作用：

主要是检查APP应用在网络不好的情况下，APP应用有无出现超时无响应、闪退、报错、消息堆积(发大量信息，堆积的话会造成顺序错乱)

③、弱网如何测试？

模拟网络情况不好，没有出现各种超时无响应、闪退、报错、消息堆积

从wifi---手机网络---网络差的情况

或者从手机网络---WiFi--网络不好情况

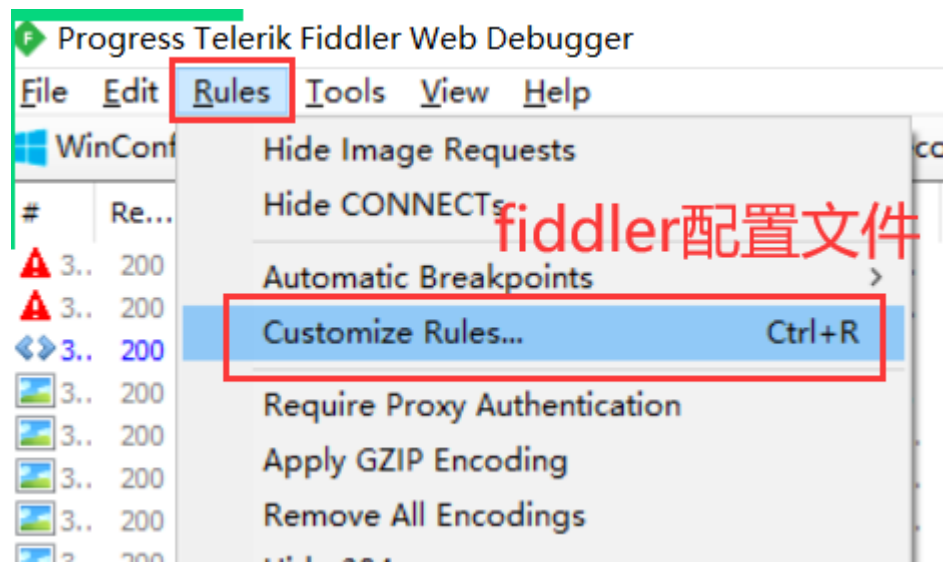
④、弱网测试的工具---fiddler

fiddler调出配置文件（快捷方式是Ctrl+R）---搜索网络上下行---并进行修改保存后关闭---勾选生效

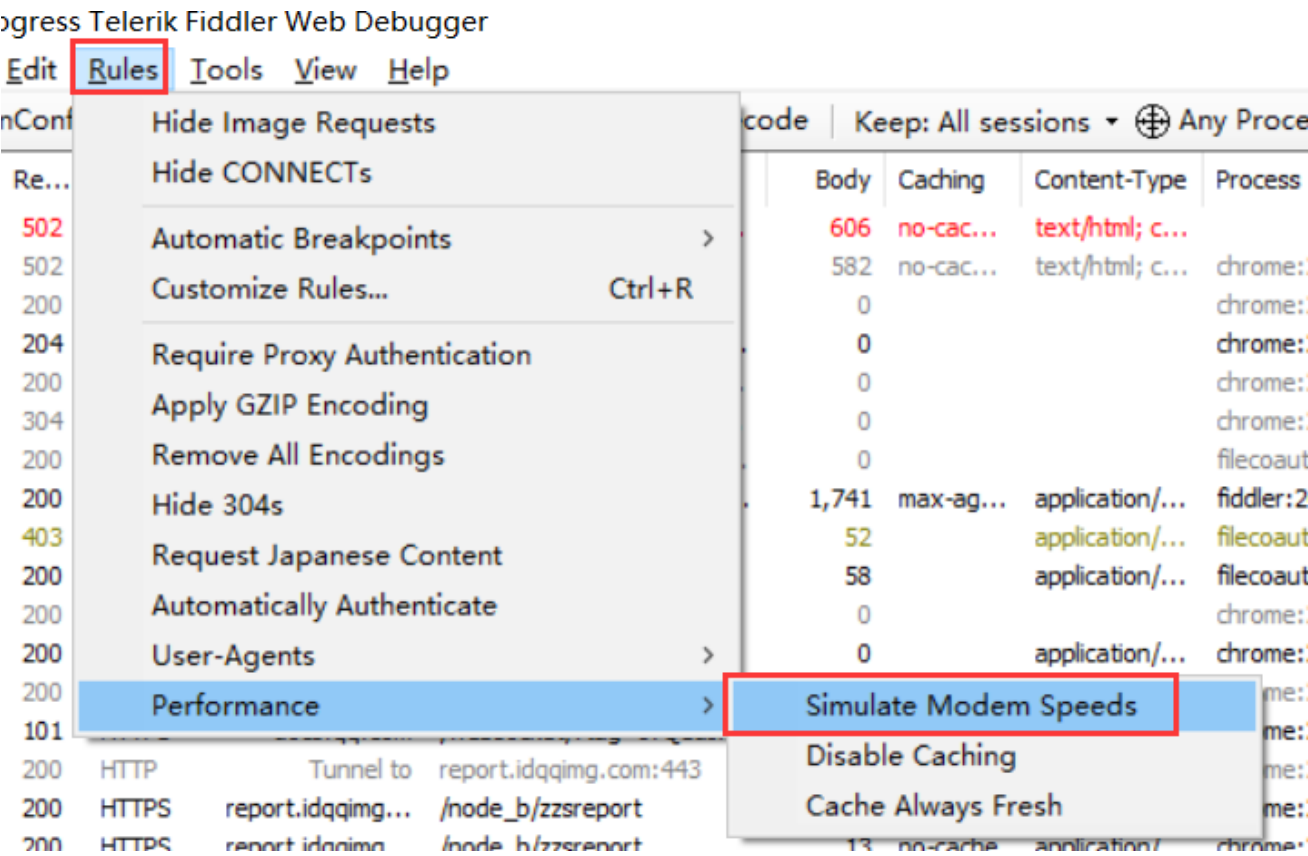
搜索：Ctrl+F 进行搜索，搜索内容：m_SimulateModem

oSession["request-trickle-delay"] = "300"; --上行速度：每延迟多少ms发送1KB请求数据

oSession["response-trickle-delay"] = "150"; --下行速度：每延迟多少ms接收1KB响应数据
(数值越小，越贴近实际网速。相反，数值越大，网络越慢)



```
if (m_SimulateModem) {  
    // Delay sends by 300ms per KB uploaded.  
    oSession["request-trickle-delay"] = "30";  
    // Delay receives by 150ms per KB downloaded.  
    oSession["response-trickle-delay"] = "15";  
}
```



5、APP权限测试

①、权限范围：

在安装或使用APP应用的过程中，会使用到手机的硬件或者软件方面的权限

权限：存储空间、摄像头、麦克风、扬声器、电话、通讯录、定位、日历、传感器、运动软件

②、权限怎么测试？

测试，主要是在检查APP应用在获取到手机相应的权限时，功能是否可以正常使用；

如果关闭了权限，那么是无法使用，而且必须要有相应的提示，告知用户如何去开启相应的权限。

6、APP交互性测试

①、交互性测试：

APP应用跟硬件、软件方面的交互

硬件：扬声器、摄像头、锁屏键、NFC、听筒、陀螺仪...

软件：相册、通讯录、天气、日历

第三方软件：QQ、微信、微博

②、交互性怎么测？

APP应用在与硬件或者软件发生交互时，需要可以正常使用，如果不能使用，那么需要有相应的提示

举例：分享视频给微信好友/朋友圈，朋友可以通过所分享的链接去查看到所分享的视频

文章、攻略、动态信息之类的都要可以分享并查看。所查看的内容是正确无误的。

7、APP兼容性测试

①、兼容性：

检查APP应用在不同手机、操作系统中，APP功能/界面是否都能正常使用，没有出现功能丧失，界面异常等情况

②、兼容性测试怎么测？

覆盖操作系统：安卓、iOS、鸿蒙

覆盖主流手机厂家：

iPhone、iOS

OPPO：ColorOS

vivo：Funtouch OS

华为、荣耀：EMUI

小米、红米：MIUI

IQOO、

三星、

一加

覆盖主流的操作系统版本：

Android：最新V12版本，需要测试版本范围：V9~V12

iOS：最新V15版本，测试版本范围：V12~V15

汇总：测试版本范围，选取的是最近三四年推出操作版本，基于手机使用寿命考虑，不一定是目标用户，不在测试考虑的范围

③、异形屏的使用

异形屏：曲面屏、水滴屏、挖空屏、刘海屏、折叠屏(贵)

异形屏的测试：需要考虑APP应用，在异形屏幕下都能适配，功能没有受到影响，可以正常使用，界面展示正常。

面试问题：兼容性测试怎么能测试到位？

第一点：公司这边会提供几台(十台八台手机)主流的手机品牌进行兼容性测试，确保主流手机品牌可以测试到位

第二点：可以使用第三方进行兼容性测试：腾讯wetest平台(要钱)、或者找第三方公司进行测试(花钱)。

第三方公司进行测试：测试top100~300手机品牌型号（项目经理沟通，测试只负责兼容性测试报告）

腾讯wetest平台：测试top100~300手机品牌型号

项目经理申请购买测试额度/负责跟第三方公司沟通兼容性测试的商务合作事宜。

测试人员只要负责使用腾讯wetest平台、或者跟第三方功能沟通兼容性测试报告即可

原型！= 需求

需求：说明交互逻辑，限制约束，可以在原型标记，也可以在旁边通过序号形式备注。

8、界面跟异常测试

界面测试：肉眼去进行判断，界面是否正常，文字有无错别字、图片、线条

字体：不能使用非授权的字体。免费使用 != 可以商用

异常测试：在促销活动（618、双十一）出现服务器繁忙，宕机，提示语方面，是否有人性化的提示（需求）：

比如：活动正在热烈促销中，请稍后再试！

9、ADB命令(做稳定性测试)

①、ADB是什么

Android debug bridge 针对安卓操作系统调试命令工具

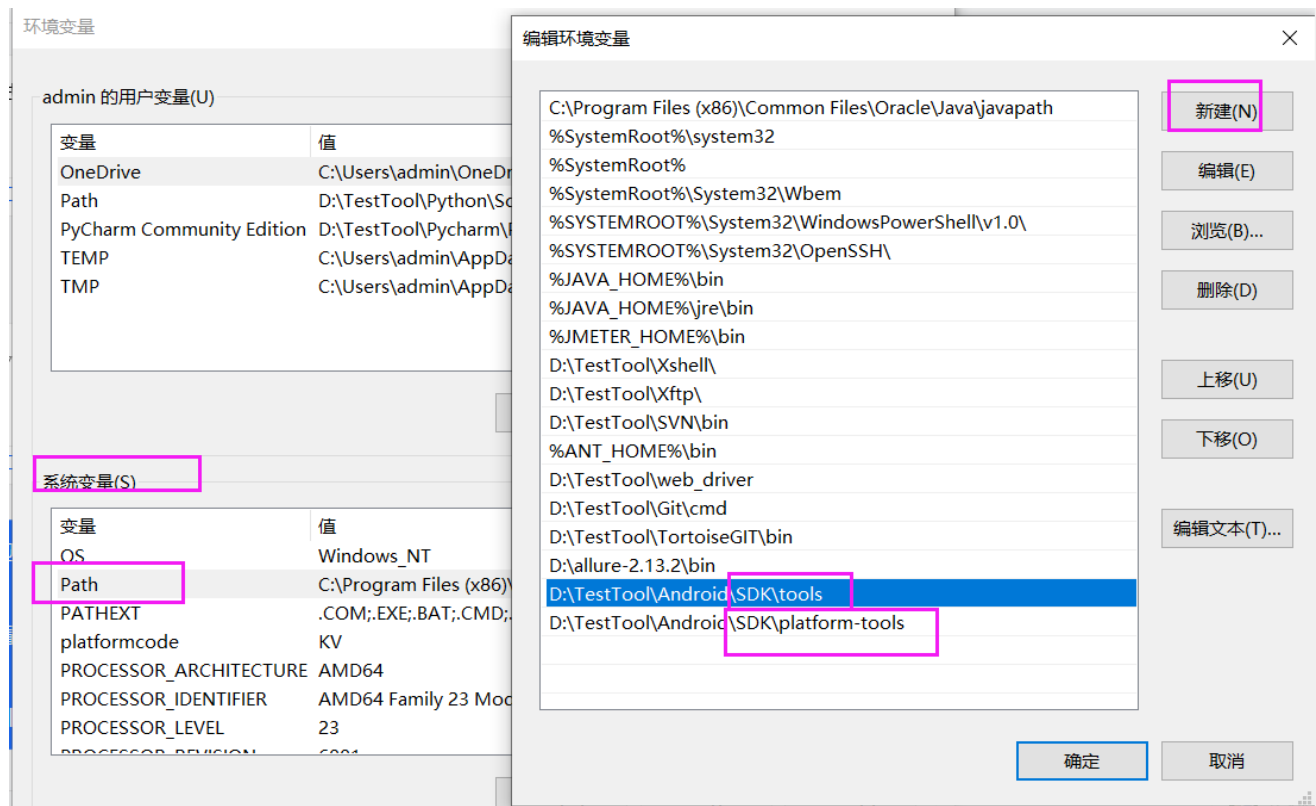
可以用来在安卓操作系统中，进行安装、卸载、获取日志、APP稳定测试、监控安卓操作系统资源！

②、安装ADB

第一步：在软件包中，将sdk.rar解压到(建议)非C盘非中文路径下

第二步：需要配置环境变量

将解压后的platform-tools、以及tools目录路径，分别添加到PATH环境变量当中



验证，在dos命令窗口下，输入：adb

```
C:\Users\admin>adb
Android Debug Bridge version 1.0.41
Version 31.0.3-7562133
Installed as D:\TestTool\Android\SDK\platform-tools\adb.exe

global options:
-a          listen on all network interfaces, not just localhost
-d          use USB device (error if multiple devices connected)
-e          use TCP/IP device (error if multiple TCP/IP devices available)
-s SERIAL   use device with given serial (overrides $ANDROID_SERIAL)
-t ID       use device with given transport id
-H          name of adb server host [default=localhost]
-P          port of adb server [default=5037]
-L SOCKET   listen on given socket for adb server [default=tcp:localhost:5
```

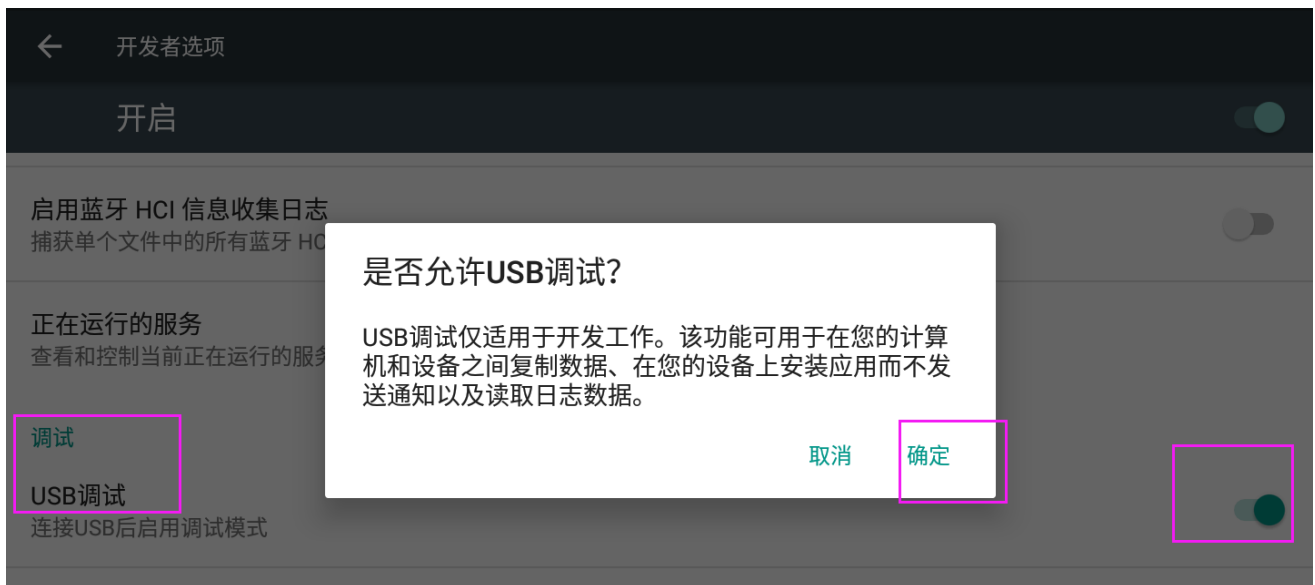
③、ADB连接安卓设备：

第一步：需要进入开发者模式，并打开USB调试：

进入开发者模式：设置---关于手机/平板电脑---版本号---连续点击7次以上

打开USB调整：设置---开发者选项---USB调试--开启





第二步：电脑通过dos命令连接设备：

1、移动设备（真机）：需要通过数据线连接，并且开启adb服务：adb start-server，才能连接上；

（关闭adb服务：adb kill-server）

```
C:\Users\admin>adb start-server
* daemon not running; starting now at tcp:5037
* daemon started successfully
```

2、模拟器：直接通过命令连接：adb connect 127.0.0.1:7555

127.0.0.1：代表本地电脑

7555：mumu模拟器的固定端口号

```
C:\Users\admin>adb connect 127.0.0.1:7555
connected to 127.0.0.1:7555
```

④、ADB命令的使用：

查看已连接设备：

adb devices


```
C:\Users\admin>adb kill-server
```

关闭adb服务

```
C:\Users\admin>sdb devices
```

'sdb' 不是内部或外部命令，也不是可运行的程序
或批处理文件。

```
C:\Users\admin>adb devices
```

```
* daemon not running; starting now at tcp:5037
```

```
* daemon started successfully
```

```
List of devices attached
```

查询已连接设备

```
C:\Users\admin>adb connect 127.0.0.1:7555
```

```
connected to 127.0.0.1:7555
```

连接mumu

```
C:\Users\admin>adb devices
```

```
List of devices attached
```

```
127.0.0.1:7555 device
```

再次查询

全新安装APP应用：

adb install 路径/文件名.apk

举例：adb install D:\fanwe.apk

(直接拖动到dos命令窗口会自动补全，

注意：建议不要使用中文路径，会出现报错)

```
C:\Users\admin>adb install D:\fanwe.apk
Performing Push Install
D:\fanwe.apk: 1 file pushed, 0 skipped. 98.6 MB/s (2826364 bytes in 0.027s)
  pkg: /data/local/tmp/fanwe.apk
Success
```

覆盖安装APP应用：

adb install -r 路径/文件名.apk

举例：adb install -r D:\fanwe.apk

```
C:\Users\admin>adb install -r D:\fanwe.apk
Performing Push Install
D:\fanwe.apk: 1 file pushed, 0 skipped. 793.6 MB/s (2826364 bytes in 0.003s)
  pkg: /data/local/tmp/fanwe.apk
Success
```

卸载APP应用

需要先知道app包名，这个包名不是文件名

查看第三方包名：adb shell pm list package -3

adb shell：adb命令固定格式

pm：安装包管理

list：列表

package -3：第三方的包（非系统自带）

```
C:\Users\admin>adb shell pm list package -3
package:com.fanwe.p2p
package:com.perfdog.app
package:com.ss.android.ugc.livelite
package:com.ss.android.article.lite
package:com.tencent.qnet
```

再通过命名进行卸载：

adb uninstall 包名

举例：adb uninstall com.fanwe.p2p

```
C:\Users\admin>adb uninstall com.fanwe.p2p
Success
```

查看安卓系统日志：

adb logcat

（清理日志：adb logcat -c）

保存日志：

adb logcat >> 路径\文件名.log

举例：adb logcat >> d:\www.log

```
1 ----- beginning of main
2
3 06-08 14:03:54.893 665 1036 D NetlinkSocketObserver: NeighborEvent(elapsedMs=3302816, 10.0.2.2, [525400123502], RTM_NEWNEIGH, NUD_STA
4
5 06-08 14:03:57.342 875 4077 D com.netease.nemu_vapi_android.thread.HttpServer: http server response data:
6 时间,月-日 时:分:秒 进程号,线程号
7 ----- beginning of system
8
9 06-08 14:03:58.420 665 693 E KernelUidCpuTimeReader: Failed to read uid_cputime: /proc/uid_cputime/show_uid_stat: open failed: ENOEN
10 (No such file or directory)
11 06-08 14:03:58.420 665 695 E KernelUidCpuTimeReader: Failed to read uid_cputime: /proc/uid_cputime/show_uid_stat: open failed: ENOEN
12 (No such file or directory)
13 06-08 14:04:02.333 665 1036 D NetlinkSocketObserver: NeighborEvent(elapsedMs=3310256, 10.0.2.2, [525400123502], RTM_NEWNEIGH,
14 NUD_REACHABLE)
```

日志级别：

V: 明细
D: 调试
I: 详细
E: 错误
W: 警告
F: 严重错误

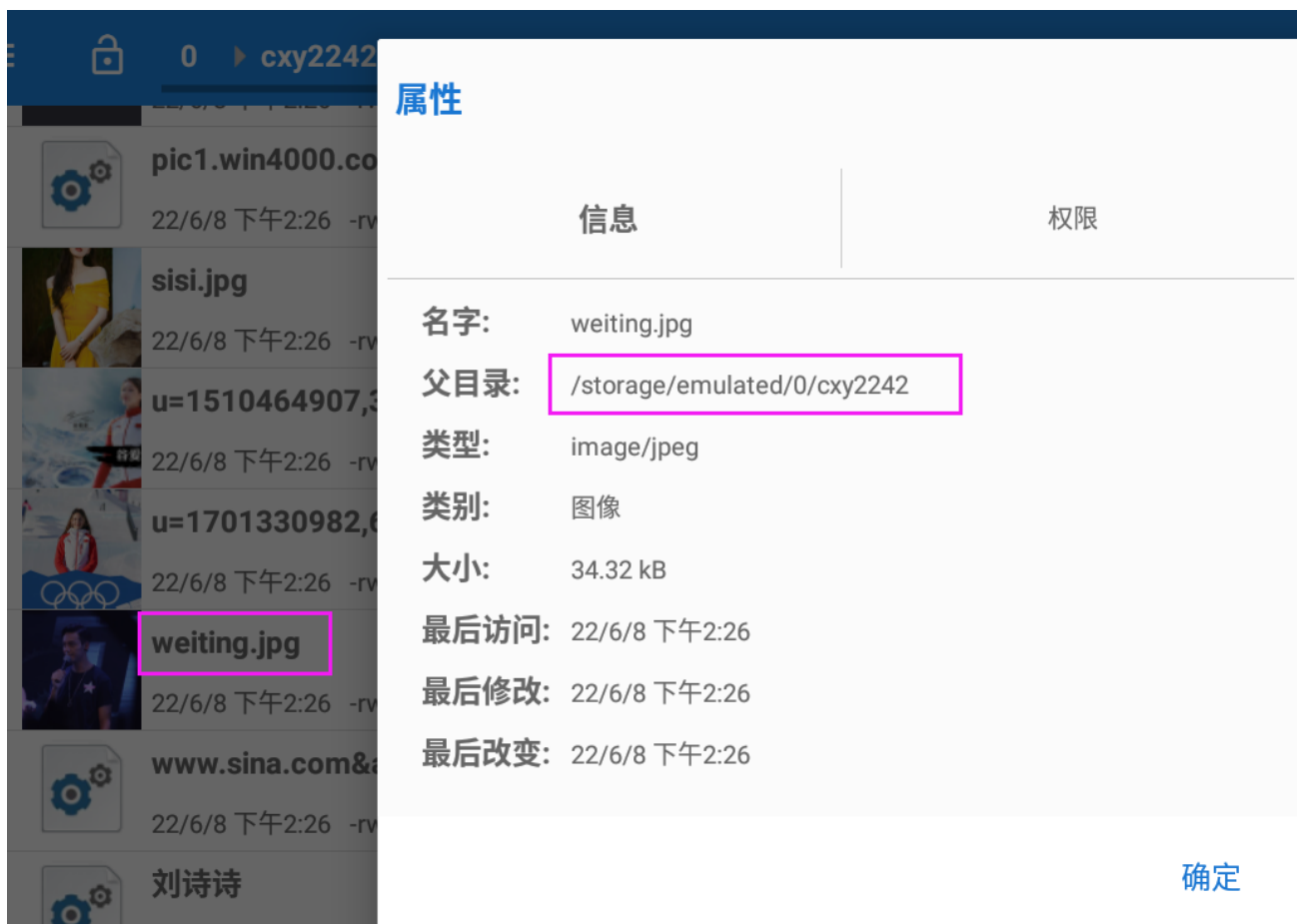
拉取文件到电脑

从安卓设备中拉取文件到电脑

第一步：需要知道文件在安卓设备中的哪个路径下

先在百度搜索图片---保存图片---去到文件管理---重命名文件为.jpg格式---会提示操作失败(实际是成功的)

将\$mumu共享文件夹重命名（以为该文件夹带特殊符号，不好使用）



可以使用 adb shell 命令进入到安卓设备中进行查看，以及复制文件路径

命令行模式: cd ls pwd exit (退出返回到电脑路径)

tab键补全

确定文件路径: /storage/emulated/0/cxy2242/weiting.jpg

第二步，通过命令直接拉取文件：

需要通过exit 命令退出返回到电脑路径

adb pull 安卓路径\文件名.拓展名 电脑路径\文件名.拓展名

(路径都不能包含中文)

举例：adb pull /storage/emulated/0/cxy2242/weiting.jpg D:\ant\weiting.jpg

```
C:\Users\admin>adb pull /storage/emulated/0/cxy2242/weiting.jpg D:\ant\weiting.jpg
/storage/emulated/0/cxy2242/weiting.jpg: 1 file pulled, 0 skipped. 3.0 MB/s (35147 bytes in 0.011s)
C:\Users\admin>
```

直接拉取文件夹

```
C:\Users\admin>adb pull /storage/emulated/0/cxy2242 D:\ant\abc
/storage/emulated/0/cxy2242/: 11 files pulled, 0 skipped. 3.0 MB/s (105470628 bytes in 33.793s)
```

推送文件到安卓设备

将电脑文件推送到安装设备中

第一步：需要确定电脑的文件，以及需要将文件存放到安卓设备中的那个路径下

电脑文件路径: E:\File\guiwu.png

安卓设备路径: /storage/emulated/0/cxy2242

第二步：直接使用adb命令推送

adb push 电脑路径\文件名.拓展名 安卓路径\文件名.拓展名

举例：adb push E:\File\guiwu.png /storage/emulated/0/cxy2242/gw.png

```
C:\Users\admin>adb push E:\File\guiwu.png /storage/emulated/0/cxy2242/gw.png
E:\File\guiwu.png: 1 file pushed, 0 skipped. 186.3 MB/s (244540 bytes in 0.001s)
```

APP稳定性测试

为什么要做稳定性测试：稳定性 == 长时间

行业发展，移动设备便携性，满足用户长时间的使用，走到哪用到哪，为了确保APP应用可以满足用户是日常使用需求，所以需要稳定性测试。

稳定性测试，就是测试APP应用在长时间(6个小时以上)使用的情况下，有无出现超时无响应、报错、崩溃的信息。

怎么进行稳定性测试？

使用 adb monkey命令进行测试

基础命令：adb shell monkey -p 包名 -v -v -v 操作次数

adb shell：adb命令基本格式

monkey: 类似猴子上蹿下跳, 不可预估的操作

-p 包名: 指定使用APP包

-v : 表示操作(步骤)记录详细级别, 一般是使用 -v -v -v显示最详细操作(步骤)记录

操作次数: 操作使用APP次数

(查询包名: adb shell pm list package -3)

举例: adb shell monkey -p com.ss.android.article.lite -v -v -v 200

(次数: 包含了有效跟无效操作, 不管是点击, 还是滑动, 又或者是触摸)

添加思考时间: adb shell monkey -p 包名 --throttle 毫秒 -v -v -v 操作次数

--throttle 毫秒: 思考时间, 每一步之间都会进行停顿思考

(停顿完成后, 如果APP应用还是没有反应, 依旧会继续操作)

举例: adb shell monkey -p com.ss.android.article.lite --throttle 500 -v -v -v 20

将adb的monkey命令操作步骤保存到日志文件:

adb shell monkey -p 包名 --throttle 毫秒 -v -v -v 操作次数 >> 路径\文件名.log

举例: adb shell monkey -p com.ss.android.article.lite --throttle 500 -v -v -v 20 >> E:\File\aa\monkey.log

```
1 :Monkey: seed=1654903255212 count=20
2           种子数           总的操作次数
3 :AllowPackage: com.ss.android.article.lite
4                               包名
5 :IncludeCategory: android.intent.category.LAUNCHER
6
7 :IncludeCategory: android.intent.category.MONKEY
8
9 // Selecting main activities from category android.intent.ca
0
```

事件比例: Event percentages

// 0: 15.0% 触摸事件TOUCH, --pct-touch

// 1: 10.0% 手势事件MOTION, --pct-motion

// 2: 2.0% 两指缩放事件PINCHZOOM,--pct-pinchzoom

// 3: 15.0% 轨迹球事件TRACKBALL,--pct-trackball

// 4: -0.0% 屏幕旋转事件ROTATION, --pct-rotation

// 5: 25.0% 基本导航事件nav, --pct-nav

// 6: 15.0% 主要导航事件majornav, --pct-majornav

```
// 7: 2.0% 系统按钮事件sysops, --pct-syskeys
// 8: 2.0% 启动activity事件appswitch, --pct-appswitch
// 9: 1.0% 键盘轻弹事件flip, --pct-flip
// 10: 13.0% 其它事件,包括按键和不常用的按键, --pct-anyevent
```

可以通过设置事件比例来控制monkey去执行操作：只要触摸点击、手势操作、两指缩放

adb shell monkey -p 包名 --throttle 毫秒 事件 比例 -v -v -v 操作次数 >> 路径\文件名.log

举例：

```
adb shell monkey -p com.ss.android.article.lite --throttle 100 --pct-touch 30 --pct-motion 20 --pct-pinchzoom
40 -v -v -v 200 >> E:\File\aa\monkey_1.log
```

注意：设置事件比例时，单个事件不宜超过50%，总的时间比例不能超100%

```
// Event percentages:
```

```
// 0: 30.0%
```

```
// 1: 20.0%
```

```
// 2: 40.0%
```

```
// 3: 2.0547945%
```

```
// 4: -0.0%
```

```
// 5: -0.0%
```

```
// 6: 3.4246576%
```

```
// 7: 2.0547945%
```

```
// 8: 0.2739726%
```

```
// 9: 0.2739726%
```

```
// 10: 0.1369863%
```

忽略错误:

稳定性测试需要将APP应用运行6~9小时以上，一般都是晚上下班后运行，可以在21/22点后运行，第二天回来看结果。

在长时间运行的过程中，如果出现报错、崩溃，那么会浪费一晚上的时间，所以需要在monkey命令中添加跳过/忽略错误。

忽略错误命令:

--ignore-crashes 忽略崩溃

--ignore-timeouts 忽略超时

--ignore-security-exceptions 忽略安全信息

--ignore-native-crashes 忽略本地代码导致的崩溃

--monitor-native-crashes 忽略监视系统中本地代码发生的崩溃

举例:

```
adb shell monkey -p com.ss.android.article.lite --throttle 500 --pct-touch 30 --pct-motion 20 --pct-pinchzoom 40 --ignore-crashes --ignore-timeouts --ignore-security-exceptions --ignore-native-crashes --monitor-native-crashes -v -v -v 64800 >> E:\File\aa\monkey_2.log
```

注意: 运行时间需要结合 思考时间跟 运行次数 来设置

思考时间: 500 0.5秒

需要运行一晚上: 9个小时

计算运行次数: $9 \times 60 \times 60 \times 2 = 64800$ 次

种子数: 复现BUG

种子数: monkey在执行操作的时候, 会将一定规律性的操作步骤设置成固定的种子数。

简单理解为。种子数是相同的, monkey命令的执行步骤一定相同的

```
adb shell monkey -p 包名 -s 种子数 --throttle 毫秒 事件 比例 -v -v -v 操作次数 >> 路径\文件名.log
```

举例:

```
adb shell monkey -p com.ss.android.article.lite -s 1654904142038 --throttle 100 --pct-touch 30 --pct-motion 20 --pct-pinchzoom 40 -v -v -v 200 >> E:\File\aa\monkey_2.log
```

在进行稳定性测试的时候, 如果运行了一晚上, 而期间出现了BUG, 那么第二天将系统日志跟monkey操作日志, 都提BUG给开发, 开发解决后, 再次使用种子数来回归测试, 去验证bug是否解决。

10、APP的性能测试

针对APP做的性能测试。

测试工具：PerfDog（腾讯旗下）收费的，**初学者新注册可以有30分钟的体验时间**

使用：需要下载，解压即可使用。针对APP做性能测试，需要通过数据线连接手机

步骤1：下载解压打开

解压后的文件					
名称	修改日期	类型	大小		
apk	2022-3-9 10:49	文件夹			
cache	2022-4-14 14:02	文件夹			
data	2022-3-9 11:00	文件夹			
jre	2021-9-1 11:26	文件夹			
libs	2022-1-18 16:30	文件夹			
log	2022-4-14 14:02	文件夹			
adb.pd	2022-4-14 14:02	PD 文件	1 KB		
bc_00000LVSQV3ARNLT_05.db	2022-4-14 11:19	Data Base File	12 KB		
conf.pd	2022-4-14 14:02	PD 文件	1 KB		
PerfDog(v6.1.220128-Win).zip	2022-1-19 08:42	压缩(zipped)文件...	159,168 KB		
PerfDog.exe	2022-1-18 16:31	应用程序	155 KB		
PerfDog.ini	2022-1-18 16:31	配置设置	1 KB		
PerfDog.jar	2022-1-18 16:31	Executable Jar File	55,642 KB		

步骤2：注册 --- 登陆 成功



需要选择连接的设备：

1、模拟器：通过adb connect 127.0.0.1:7555连接

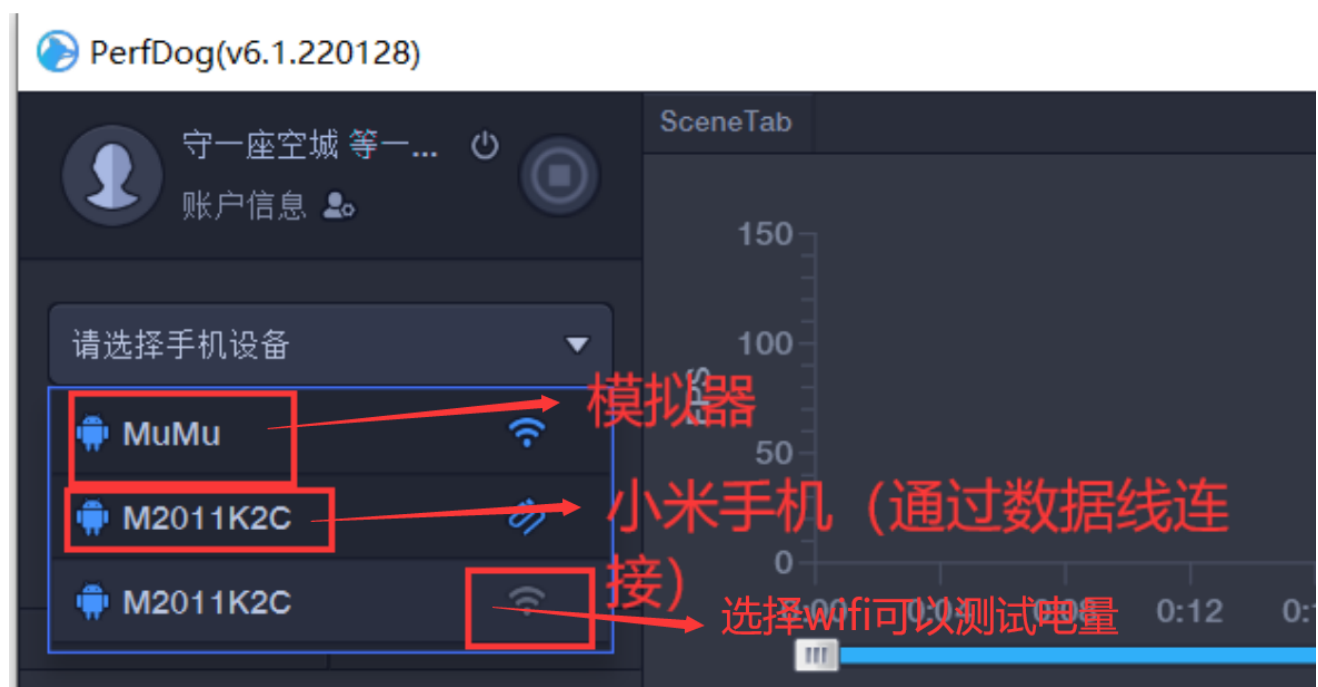
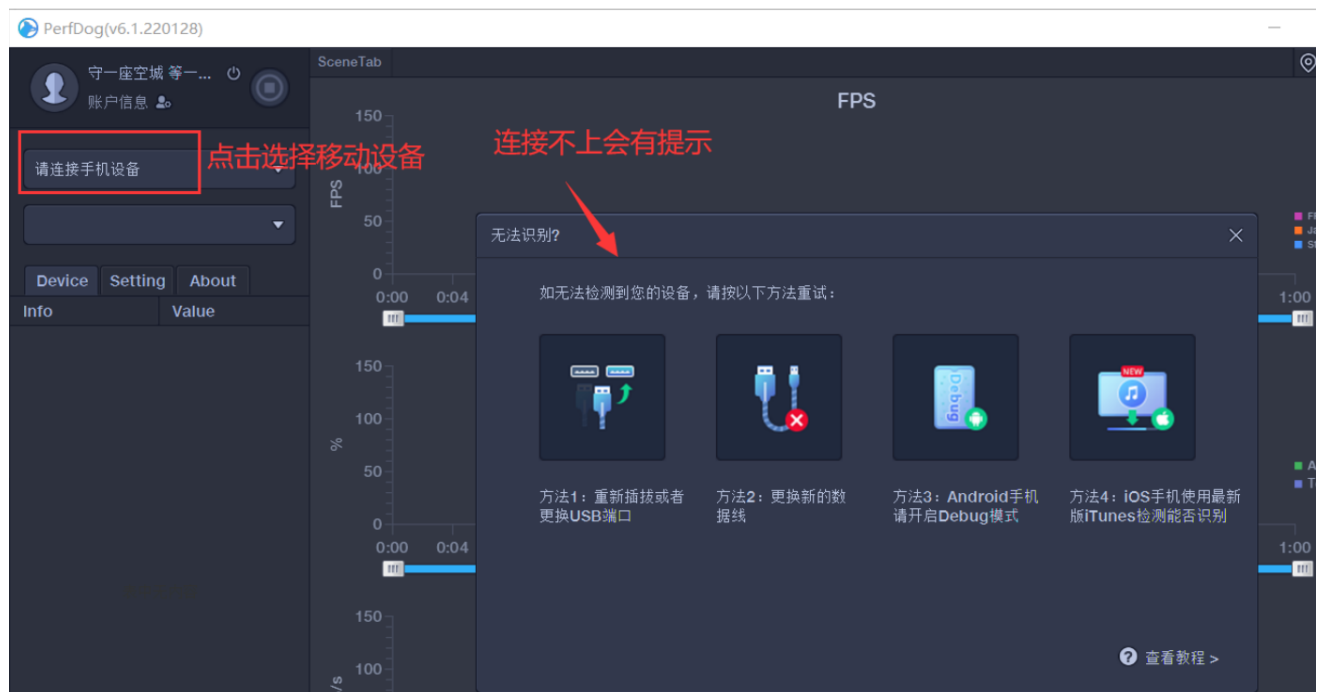
2、真机：adb start-server ,并通过数据线连接

注意：需要基于账户 有使用额度的情况下，才能继续选择设备

如果电脑没有连接好移动设备，会有相应的提示：

- ①真机需要进入开发者模式、同时打开USB调试
- ②数据线问题、USB插口问题（部分劣质数据线只有充电功能，没有传输文件数据功能）

步骤3、选择设备



步骤4：选择已安装的APP应用，来进行性能测试

【【选择应用后会自动开始性能测试监控】】



守一座空城 等一...



SceneTab

账户信息



M2011K2C



请选择测试应用

查找应用

选择移动设备上已安装的应用

已安装的应用



360借条



58同城

FPS

150

100

50

0

0

150

100

%



步骤5：查看APP性能监控指标

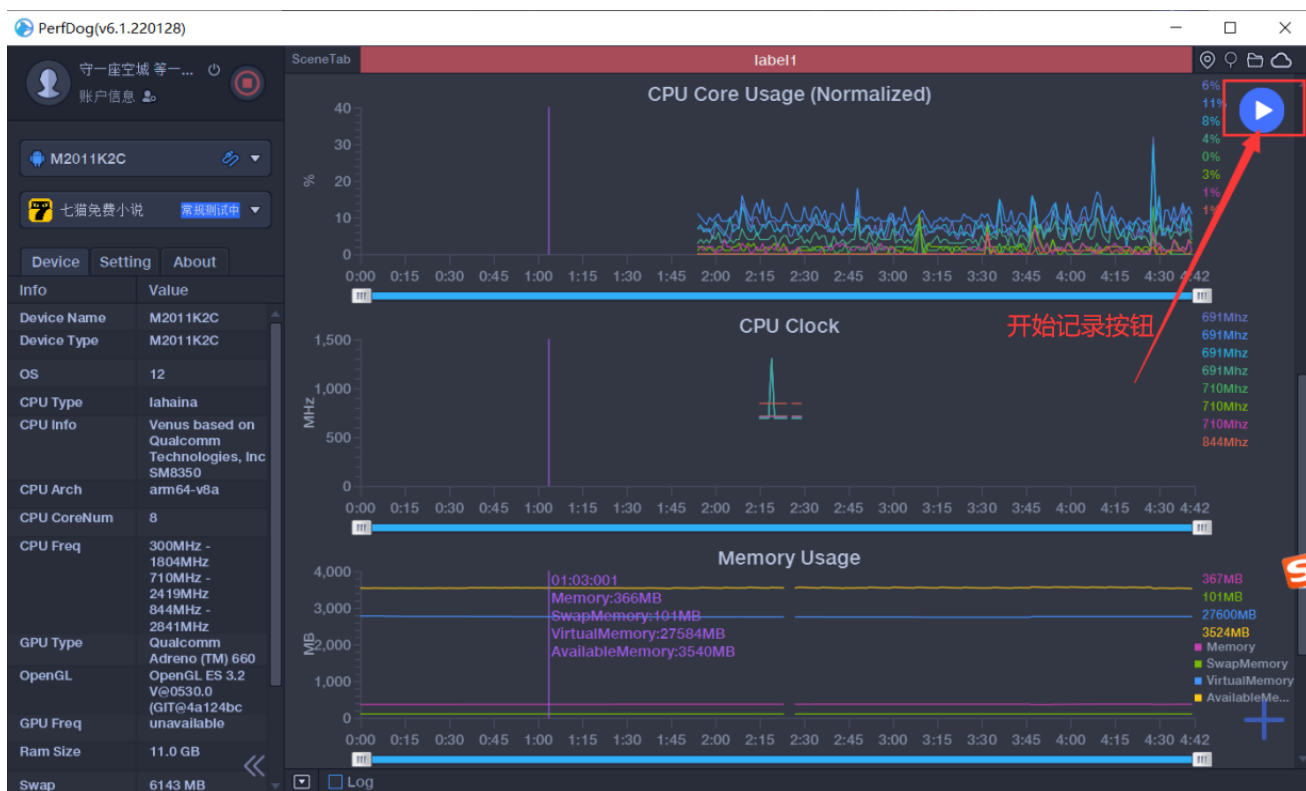
注意通过右下角的加号 + 来增加需要监控的性能指标。



步骤6、开始记录性能测试数据

前面可以先把APP运行起来，先热机（热身运动，1~2分钟不做监控）

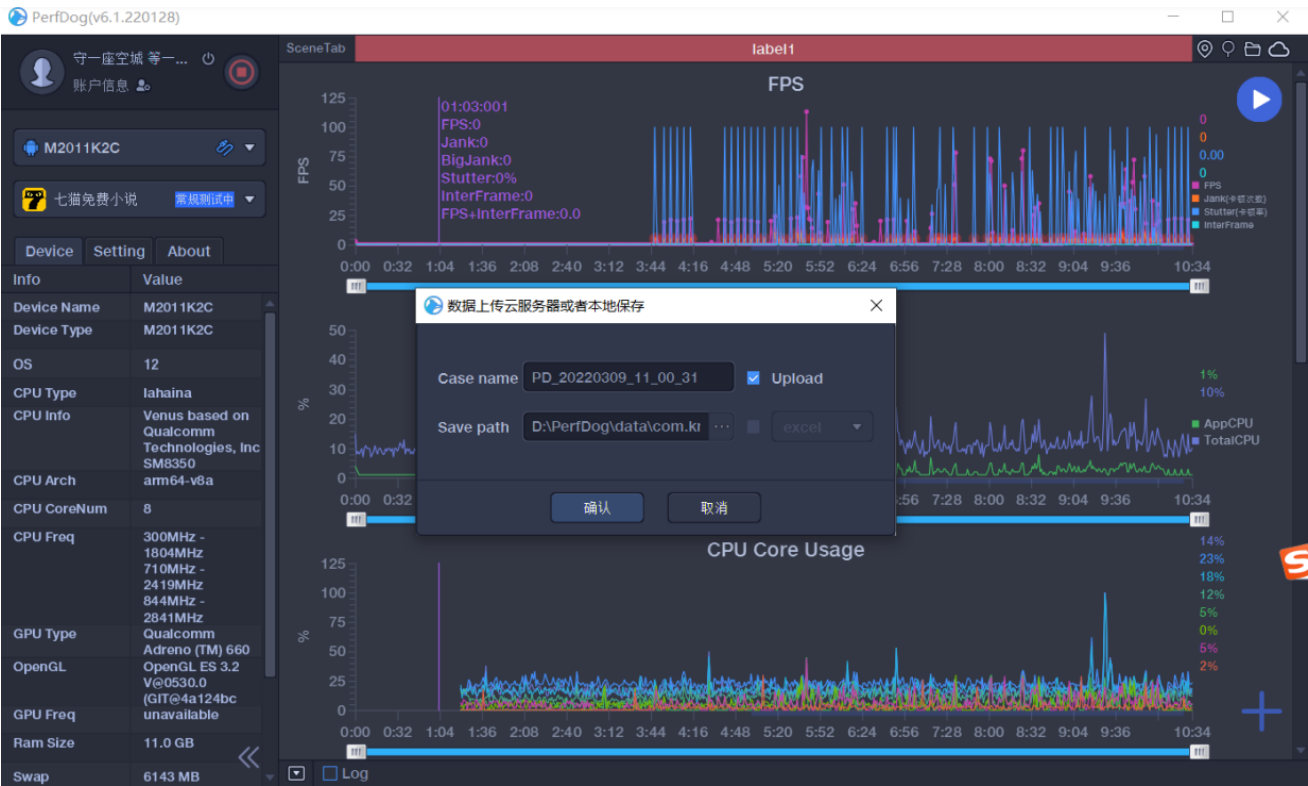
点击右上角的三角形按钮，开始记录性能测试数据，记录信息



步骤7、测试完成，记录并保存数据

每次APP性能测试时间在15分钟~30分钟。再次点击右上角的开始记录按钮，进行结束测试，并保存性能测试数据
点击右上角的开始/停止按钮，结束性能测试监控：

同时会弹出保存数据窗口， **务必记得勾选，将性能数据上传性能狗平台**



步骤8：查看APP的性能指标参数

性能指标：

FPS：帧率，60、120、144

帧率代表流畅程度，但是帧率越高，耗电量越快。

CPU：处理器

使用率月底，APP性能越好。但是部分应用(游戏/视频)对CUP使用是很大的
只要APP对CPU使用消耗并不是特别大(七八十以上)，基本不用关注。

Memory：内存，不超过手机使用运行内存即可

GPU：图形处理器，使用率不宜过高

NetWork：流量，考虑在相同时间范围内，流量消耗越低越好

Batter：电量，考虑在相同时间范围内，耗电量消耗越低越好

最后面进行APP性能指标的判断：

第一种：纵向对比：拿历史版本性能指标数据来对比，V 1.3版本-->V2.0版本，判断下大版本升级，对APP方面的性能有没有太大变化。变化太大，判断下是否因为功能的增加而导致性能指标的上升，是否在预期范围内。是否可以接受。

（好比是否因为环境/条件的影响，导致你饭量比之前吃得多，使用得多）

第二种：横向对比：拿市面上同类型的应用产品来对比。

阅读类：视频类、游戏类；

CPU：23%，---- 26%/41% 26是正常的，41是可以跟开发/产品沟通，提出疑问，希望优化。

其他性能指标，也是同理，跟同类型的产品进行对比。

（好比学习成绩，别人家的小孩学习是怎样的，我也要达到这样的水平）

六、APP测试面试问题（知识点---知识面）

1、APP测试跟Web的测试有什么不同？（重点）

app跟web都是以功能测试为主。

而web项目的兼容性测试，主要体现在浏览器（五大内核浏览器）跟操作系统（微软window、苹果电脑）

APP项目兼容性测试，主要体现手机的操作系统、操作系统的版本、手机厂家、手机型号，有些时候还需要考虑到异形屏对APP的功能有无造成影响。

除了兼容性，APP还需要做安装卸载测试、权限测试、交互性测试、稳定性测试、APP的性能测试。

每个专项测试要知道具体怎么做

2、（APP或web）项目测试流程（重点）

需求大会---测试计划---测试用例编写---测试用例评审---执行测试用例---提BUG---回归验证---出测试报告---上线

3、fiddler抓取APP中的HTTPS请求（重点）

前提条件：在同一局域网内，查找电脑的IP地址

fiddler方面的操作：开启远程代理，设置默认端口号；https的请求---还需勾选开启可以抓取https请求(安装证书)

设备中的操作：修改WiFi成手动代理，代理服务器设置成电脑的IP地址，端口号为fiddler远程代理的端口号；

https的请求---设备需要下载证书（浏览器访问电脑IP+fiddler端口号），并进行安装。

4、常用的ADB命令（次重点）

查询已连接的设备：adb devices

开启、关闭服务：adb start-server 、 adb kill-server

安装应用：adb install 路径\文件名.apk

查询已安装应用包名：adb shell pm list package -3

卸载应用：adb uninstall 应用包名

查看并保存日志：adb logcat >> 文件名.log

传输文件：拉取文件 adb pull 设备路径\文件名.拓展名 电脑路径\文件名.拓展名

推送文件 adb push 电脑路径\文件名.拓展名 设备路径\文件名.拓展名

5、稳定性测试（次重点）

之前做稳定性测试，是需要运行8~10个小时，那么是在晚上下班后，一般在晚上20点/21点之后，挑选同事基本都下班了，对测试影响最小。

使用的monkey命令：adb shell monkey -p 包名 -v -v -v 次数，

一般为了模拟真实用户的点击操作，我们需要加上思考时间，设置的思考时间是500毫秒，那么为了长时间运行9个小时，设置的次数在6~7万次。

我们还是针对实际的页面场景设置事务比例，为了避免再长时间稳定性测试过重出现报错或崩溃等原因影响测试过程，还可以增加：忽略崩溃、忽略超时、忽略安全信息、忽略本地代码导致的崩溃、忽略监视系统中本地代码发生的崩溃。

在第二天回来的时候去查看系统日志，查找有无F、E、W报错信息，有的话，将日志提交给开发进行修复。

修复完成后，在晚上再使用种子数进行回归验证。

6、APP性能测试（一般）

之前大版本的更新迭代有做APP的性能测试，是使用性能狗工具（公司购买充值的账号）来做的。小版本的更新没有做。

做性能测试：

之前是有一位同事/之前我这边也有做过，但做得不多，说下性能测试的步骤、需要查看对比的性能指标

步骤1：下载解压打开

步骤2：注册 --- 登陆 成功

步骤3、选择设备

步骤4：选择已安装的APP应用，来进行性能测试

步骤5：查看APP性能监控指标

步骤6、开始记录性能测试数据

步骤7、测试完成，记录并保存数据

步骤8：查看APP的性能指标参数

性能指标：

FPS：帧率，60、120、144

帧率代表流畅程度，但是帧率越高，耗电量越快。

CPU：处理器

使用率月底，APP性能越好。但是部分应用(游戏/视频)对CUP使用是很大的

只要APP对CPU使用消耗并不是特别大(七八十以上)，基本不用关注。

Memory：内存，不超过手机使用运行内存即可

GPU：图形处理器，使用率不宜过高

NetWork：流量 考虑在相同时间范围内，流量消耗越低越好

Batter：电量，考虑在相同时间范围内，耗电量消耗越低越好

最后面进行APP性能指标的判断：

第一种：纵向对比：拿历史版本性能指标数据来对比 V 1.3版本-->V2.0版本，判断下大版本升级，对APP方面的性能有没有太大变化。变化太大，判断下是否因为功能的增加而导致性能指标的上升，是否在预期范围内。是否可以接受。

(好比是否因为环境/条件的影响，导致你饭量比之前吃得多，使用得多)

第二种：横向对比：拿市面上同类型的应用产品来对比。

