

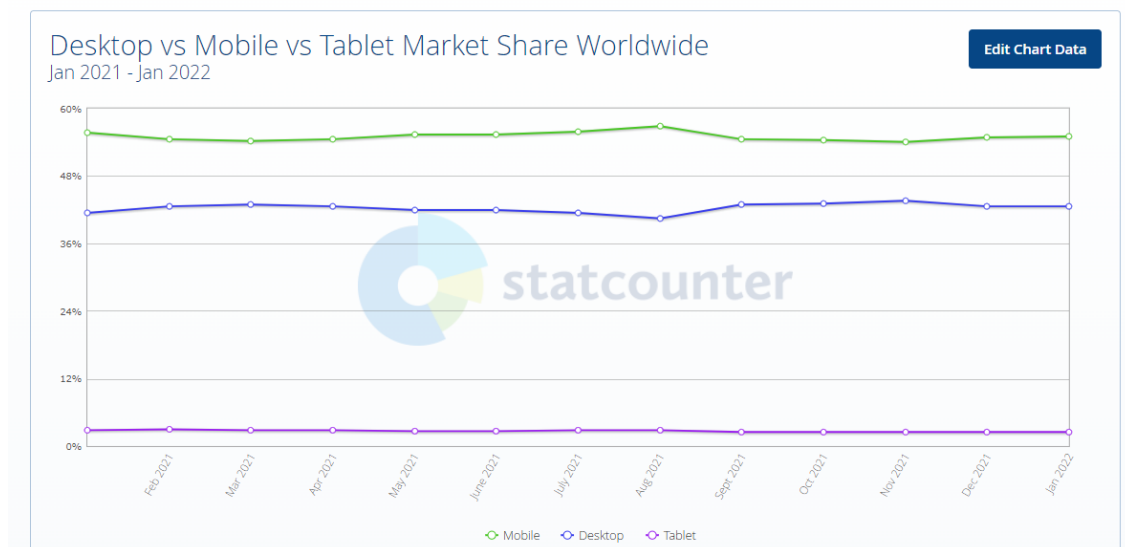
App测试

1. 什么是App?

- App是Application的缩写，意思是“应用程序”，主要指安装在智能手机、智能平板、车载智能终端及可穿戴设备等终端上的软件，比如我们常用的手机版QQ、微信、抖音等。



- Mobile端和PC端流量统计数据:



<https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet>

- 2021年1月中国手机App用户量排行榜Top50:

排名	APP名称	1月活跃人数(万人)	月活跃用户 环比增幅
1	 微信	97,563.8	0.51%
2	 QQ	76,595.9	0.33%
3	 手机淘宝	70,628.0	2.76%
4	 支付宝	70,286.5	1.01%
5	 爱奇艺	62,936.6	2.67%
6	 抖音短视频	56,766.8	5.85%
7	 腾讯视频	53,846.5	2.58%
8	 优酷视频	48,537.8	3.85%
9	 快手	44,725.9	3.35%
10	 拼多多	42,836.7	-3.38%
11	 高德地图	41,841.8	-0.83%
12	 百度地图	38,724.7	-3.01%
13	 百度	38,106.9	7.02%
14	 新浪微博	37,892.9	1.28%
15	 搜狗输入法	36,586.3	0.29%
16	 今日头条	32,682.2	5.13%
17	 WiFi万能钥匙	32,594.4	-0.41%
18	 QQ浏览器-便捷管理...	29,254.2	-0.18%
19	 腾讯新闻	29,183.8	7.63%
20	 腾讯手机管家	28,182.2	-1.17%
21	 应用宝	27,516.1	-0.37%
22	 京东	26,781.0	-0.74%
23	 酷狗音乐	26,718.7	-3.63%
24	 QQ音乐	26,087.2	-3.91%
25	 UC浏览器	21,680.7	7.71%
26	 华为应用市场	17,338.4	2.69%
27	 全民K歌	16,180.7	2.38%
28	 360手机卫士	15,161.9	-0.62%
29	 抖音火山版	14,893.6	4.66%
30	 西瓜视频	14,189.1	-0.89%
31	 芒果TV	13,583.2	6.75%
32	 360清理大师	11,986.6	-0.39%
33	 讯飞输入法	11,948.4	0.21%
34	 墨迹天气	11,544.7	2.01%
35	 OPPO软件商店	11,395.0	-2.97%
36	 酷我音乐	11,351.6	-0.84%

37		360手机助手	11,086.9	-0.19%
38		美团	10,686.9	-5.50%
39		小红书	10,662.4	0.54%
40		美图秀秀	10,009.7	7.19%
41		哔哩哔哩	9,742.1	16.27%
42		好看视频	9,084.3	3.30%
43		网易云音乐	9,026.7	-0.25%
44		趣头条	8,966.0	-2.82%
45		百度手机助手	8,757.5	-0.21%
46		喜马拉雅	8,684.4	-5.31%
47		WiFi管家	8,611.3	-1.52%
48		小米应用商店	8,607.5	4.56%
49		vivo应用商店	8,519.9	-0.67%
50		WPS Office	8,414.4	0.75%

- 为什么要学习App测试？
 - 当今是移动互联网时代，据统计，每月移动互联网用户规模达到10亿+人，App用户人均单日使用时长达到6小时+，排除吃饭睡觉，有一多半的时间在玩手机。鉴于此，软件开发者在开发软件时，除了Web端应用外，大多数时候还会提供对应的App端应用。作为测试工程师，必须掌握App测试的相关技能。

2. App的分类

- iOS是基于Unix内核深度定制的一个移动设备操作系统；Android是一个Linux发行版，用于移动设备；Linux是类Unix系统；iOS是闭源的，Android是开源的；iOS是苹果公司主导，Android是谷歌主导的；
- 从操作系统平台来分，主要分为如下两种类型：
 - iOS App
 - 安装在苹果的iOS操作系统上，主要是iPhone、iPad、iWatch等
 - 同一款应用，通常对iPhone和iPad要出两个不同的版本，因为要针对iPhone和iPad两种终端的不同分辨率、屏幕尺寸分别适配
 - iOS App的安装包后缀名为.ipa（要记忆）
 - Android App
 - 安装在Android操作系统上，主要是Android手机和Android平板
 - 同一款应用，通常对Android手机和Android平板要出两个不同的版本，因为要针对Android手机和Android平板两种终端的不同分辨率、屏幕尺寸分别适配
 - Android App的安装包后缀名为.apk
- 从技术实现上，目前主流的App可以分为原生App、混合App和Web App三种类型：
 - 原生App
 - 原生App又称Native App，就是利用Android、iOS平台官方的开发语言、开发类和库以及工具进行开发。比如安卓的Java语言，iOS的Object-c语言。在应用性能上和交互体验上非常不错。原生App需要针对Android和iOS平台开发两套不同代码，成本高。（原生APP最大的优势就是性能，是三种APP开发模式里面最好的一个）
 - Web App

- 指的是移动端的Web浏览器，主要借助H5网页技术进行开发，相当于开发一个网页，然后分别生成Android和iOS的壳。这种App只需要一套编程开发语言，技术简单，人力少，周期短。缺点是用户体验相对差。
- 混合App
 - 混合App又称Hybrid App，是介于原生App和Web App之间的一种App形式。
 - uni-app是一个使用 Vue.js 开发所有前端应用的框架，开发者编写一套代码，可发布到iOS、Android、H5、以及各种小程序（微信/支付宝/百度/头条/QQ/钉钉）等14个平台。
 - 了解uni-app: <https://uniapp.dcloud.io/README>

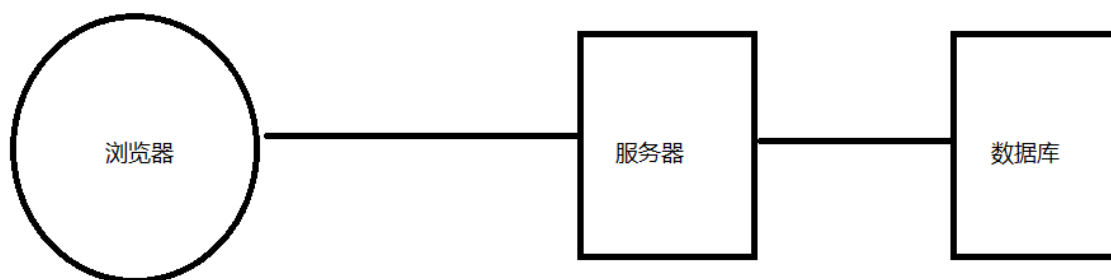
目前国内基本上都是以混合app为主。

3. App应用商店

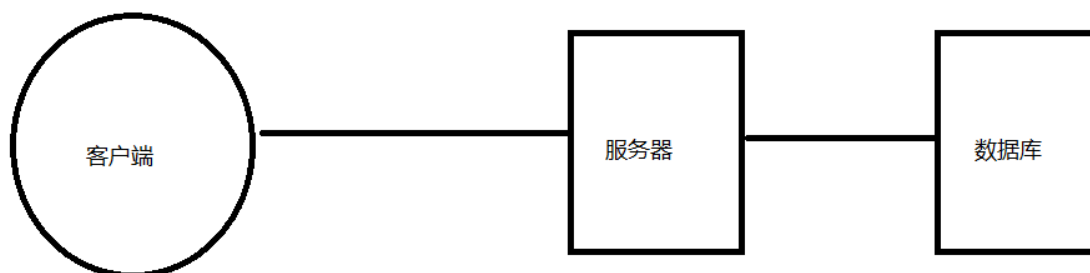
- 常用的App应用商店：
 - App Store：苹果应用商店，所有非越狱的苹果设备，只能从App Store上下载App
 - Google Play：谷歌应用商店，国外的智能手机默认使用的应用商店
 - App Gallery：华为应用商店，华为手机和平板使用的应用商店
 - 小米商店应用：小米手机和平板使用的应用商店
 - OPPO软件商店：OPPO手机使用的应用商店
 - 百度手机助手、360手机助手、腾讯应用宝、豌豆荚等其它应用商店，中国有超过400个Android应用商店。
- 注意：凡是在以上商店能下载的App都是正式版本，测试版本不可能从上面去下载。

4. B/S架构与C/S架构

B/S架构：



C/S架构：



1. B/S：浏览器/服务器，用浏览器去打开的系统；C/S：客户端/服务器，客户端又分手机客户端（App）以及PC客户端
2. B/S架构下核心的业务处理在服务端完成，C/S架构下可以将业务合理地分配到客户端和服务端来实现

3. C/S能更好地利用客户端设备的处理能力，具有更好的用户体验，响应速度更快，安全性也更高
4. 在开发、测试时，B/S需要考虑兼容不同的浏览器（内核），C/S需要兼容不同的操作系统版本及设备，前者相对开发周期更短，共享性更强；因为iOS和Android系统的差异，iOS App和Android App的开发语言是不一样的，对开发人员的能力要求也是不一样的，需要不同的开发人员分别开发出iOS App和Android App两套开发包，同时因为Android系统的碎片化严重，Android App在开发测试阶段需要花费大量的精力适配不同厂家的系统及设备，最后因为手机和平板的尺寸不一样，还需要分别出手机版和平板版（HD）两个版本。
 - 所有浏览器，按内核分，有如下几种内核：
 - Apple Webkit（谷歌内核）：**Chrome**、**Safari**、**Edge**、QQ浏览器、360极速版、世界之窗等
 - Gecko（火狐内核）：**Firefox**
 - Trident（IE内核）：edge、360兼容版
5. 从维护的角度，C/S工作量会更大，因为B/S的升级只需要在服务器上升级即可，C/S的升级往往需要同时升级服务器端和客户端；iOS和Android系统有重大更新的时候会需要开发者同步更新App以适配最新的操作系统。

5. iOS App与Android App的区别

1. 开发语言（原生应用）：iOS App用object-c开发，对应的是iOS开发工程师；Android App用Java开发，对应的是Android开发工程师；
2. 上架（商用上线）的市场：iOS App是统一上架到苹果的App Store，苹果设备只要没有越狱，只能从AppStore上下载app安装；Android app是上架到各个手机厂家的应用市场上，比如谷歌的Google Play，华为的App Galary等等；
3. 在做兼容性测试时，Android App会更复杂；
4. iOS是一个相对封闭的操作系统，是苹果基于Unix内核深度定制的一个移动设备操作系统；Android是一个开放的操作系统，谷歌会定期发布基线版本，各个手机厂家会在基线版本的基础上再进行本地化定制，结果就导致Android系统出现很多分支（碎片化），导致Android App对不同分支版本的适配难度会更大；
5. AppStore对应用的审核会更严格，所以iOS APP会相对更安全；
6. Android App的包后缀是.apk，iOS App的包后缀是.ipa；
 - MyCamera.apk：Android手机版App
 - MyCameraHD.apk：Android平板App
 - MyCamera.ipa：iPhone手机版App
 - MyCameraHD.ipa：iPad版App

6. App的测试点

6.1 功能测试（Web和App都要重点考虑的）

就功能测试而言，App测试和Web测试没有区别，都是基于软件的功能性需求编写测试用例，设计测试用例的方法都可以采用场景分析、等价类、边界值、错误猜测等。写好用例之后先安装环境，然后执行用例，对发现的问题提交Bug单跟进解决。

6.2 兼容性测试（Web和App的侧重点不一样）

兼容性测试顾名思义就是，要确保App在不同场景下的正确性。App和Web的兼容性测试侧重点不一样，Web需要兼容不同内核的浏览器（Chrome、Firefox、edge等），App需要兼容不同版本的操作系统（iOS14、iOS15、Android10、Android11、Android12、鸿蒙OS等）、设备尺寸、屏幕的类型（曲面屏、折叠屏、刘海屏、挖孔屏、全面屏）、分辨率等。

测试方法：在不同的设备上安装、启动、运行、卸载测试，需要大量的测试设备才能做好兼容性测试。

兼容测试的难点：设备类型太多，iPhone目前有20多款设备，Android手机就更多了，Android碎片化严重，想要做充分的兼容性测试，成本很大。

测试设备来源：公司自己买测试机、公司租手机、用员工的手机、WeTest云测平台等。一般采用真机来测试，如果实在找不到真机，有的时候也用模拟器来模拟。

接下来就是将app的专项测试：以下几个大点

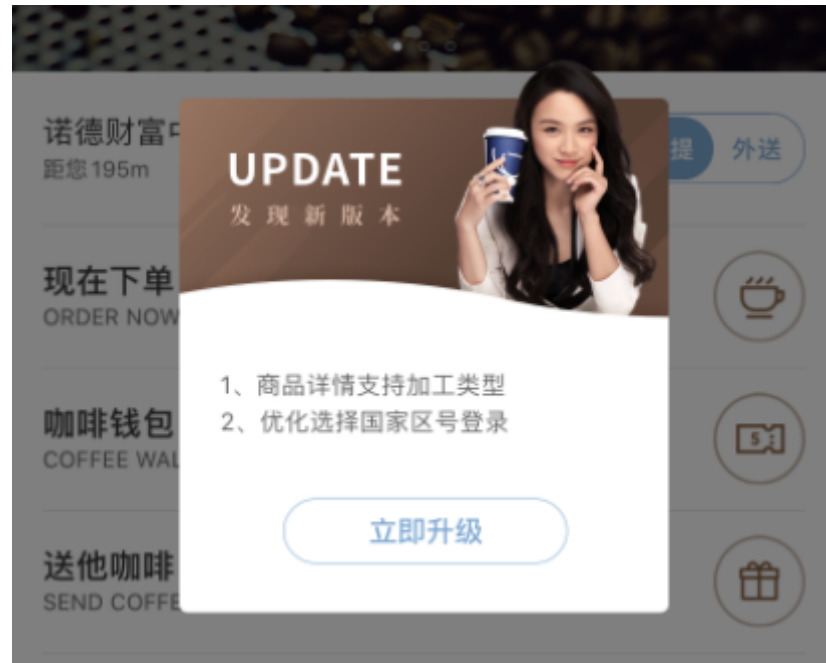
6.3 安装、卸载、升级测试

- 安装测试
 - 安装方法：
 1. 对Android App来讲，可以直接把apk发到手机上下载并安装；
 2. 通过adb（安卓调试桥）命令安装（仅限于Android App）：`adb install xxx.apk`；
 3. 通过助手等工具安装（iOS App和Android App都可以）；
 - 测试点：直接安装能否安装成功，卸载后再次安装能够成功，安装的App能否正常启动
 - 注意事项：
 1. 通过助手工具连接iOS设备前，需要先在电脑上安装iTunes(爱思助手)工具
 2. 安装iOS App，一般需要先获取到测试手机的UDID（每个手机都不一样），由开发写入到App的配置文件中再打包，这时得到的App才可以在测试设备上安装
 - 卸载方法：
 1. 在手机上直接卸载
 2. 通过adb命令卸载：`adb uninstall 包名`
 3. 通过助手工具卸载
 - 测试点：App能够卸载成功，卸载后是否可以再次安装
- 升级测试
 - 基本方法：先安装老版本，再安装新版本，看是否能安装成功并覆盖老版本。
 - App的升级方案

App的升级方案有在线升级和离线升级两种，其中在线升级又分为强制升级和非强制升级。

 1. 在线升级：当应用商店已经有新版本时，提示更新，用户点击更新，下载最新版本，进行安装升级，这种就是在线升级(前提是要有网络)；
 - 强制升级：强制更新就是当检测到有新版本时，弹出提示，这个提示框没有关闭按钮，只能更新，一般需要强制更新的版本，主要是功能模板改动较大，或业务比较

重要，和功能模板存在风险等；



- 非强制升级：非强制更新就可以让用户选择是否现在更新或以后更新，旧版本的功

能正常使用。



- 2. 离线升级：不从线上下载升级包（先下载好包），通过线下的方式安装升级包进行升级，这种就是离线升级（已有升级包，安装升级包进行升级，这种就是离线升级）。

- 升级场景设计：

- 第一个版本：用户只有新安装的场景
- 第二个版本：用户有新安装的场景、也有从第一个版本升级到第二个版本的场景
- 第三个版本：用户有新安装的场景、也有从第一个版本升级到第三个版本的场景、也有从第二个版本升级到第三个版本的场景
- 版本越多，跨版本升级的场景也越复杂，一般也不可能把所有场景都验证一遍，通常的做法是分析生产环境的数据，看绝大部分用户都在哪几个版本上，重点考虑从这几个版本升级到新版本的场景。如果App做了强制升级的方案，这种情况下用户的版本就比较新，升级的场景也更简单。

- 测试方法：

1. 通过手机助手软件或者命令，先安装升级前的老版本，再安装新版本，看能否正常地覆盖升级，以及升级后功能是否正常，数据是否有丢失，版本号是否正确；
2. 为尽可能模拟真实的用户场景，可以把升级包部署在一个http服务器上（升级服务器），手机通过浏览器打开这个服务器的URL下载新版本，模拟用户从应用商店下载升级的过程。

6.4 弱网测试

一方面App的网络环境复杂，包括2G/3G/4G/5G/WiFi等，另一方面App的使用场景多变，用户可能在各种场所使用App，包括网络较差的场景（延迟、丢包、抖动等），如果App没有对各种网络异常进行兼容处理，那么用户可能在使用中遇到APP闪退、ANR（应用无响应）、数据丢失等问题。

测试方法：通过工具模拟弱网的场景进行测试，比如QNET、fiddler。

6.5 App压力（稳定性）测试

Android App可以通过monkey命令来做压力测试。

6.6 UI易用性测试（web和app都有的）

- 检测界面是否美观，操作是否简单易用，是否符合用户的操作行为，产品是否有相关的帮助文档，FAQ（常见问题）等
- 参考：UI设计图，检查系统的实现是否与UI设计图一致。

6.7 交叉事件测试（中断测试）

交叉事件测试也叫中断测试，是指App执行过程中，有其它事件或者应用中断当前应用执行的测试。比如在App使用过程中出现来电、来短信等消息、响闹钟、弹出电量低、打开其它App等其它事件，验证当其它事件出现的时候，是否会中断当前App的事件。

交叉测试一般采用手工测试的方式在真机上进行。

6.8 耗电量测试

- 目的就是检查App的耗电是否过高，电量消耗跟CPU有关系。
- 可以通过手机的电池统计来查看App的耗电情况，也可以通过adb命令来查看，还可以通过第三方工具PerfDog（性能狗）、GT等来检查耗电情况

6.9 流量测试

- 流量测试，指的是监控App的流量使用是否合理，目的是优化App的流量使用，比如有些数据需要缓存到客户端上来避免过多地消耗流量。
- 有很多方法可以获取App的流量，比如对于Android系统，网络流量信息通常存放在/proc/net/dev目录下，可以直接利用adb工具获取实时的流量信息。另外，推荐一款Android的轻量级性能监控小工具Emmagee。对于iOS系统，可以使用XCode自带的性能分析工具集中的Network Activity，分析具体的流量使用情况。

- 推荐：使用PerfDog（性能狗）可以很方便地监控App的流量。

6.10 边界测试

边界测试是指，移动App在一些临界状态下的行为功能是否正常，基本思想是找出各种潜在的临界场景，并对每一类临界场景做验证和测试。比如：

- 手机提示内存空间不足
- 飞行模式来回切换的场景
- App 不具有某些系统访问权限的场景，比如 App 由于隐私设置不能访问相册或者通讯录等
- 长时间使用App，系统资源是否有异常，比如内存泄漏、过多的链接数等
- 出现ANR的场景（网络差、操作频繁、手机上起的应用过多、不同App之间切换等）
- 操作系统时间早于或者晚于标准时间的场景
- 时区切换的场景 GMT
- 等等.....

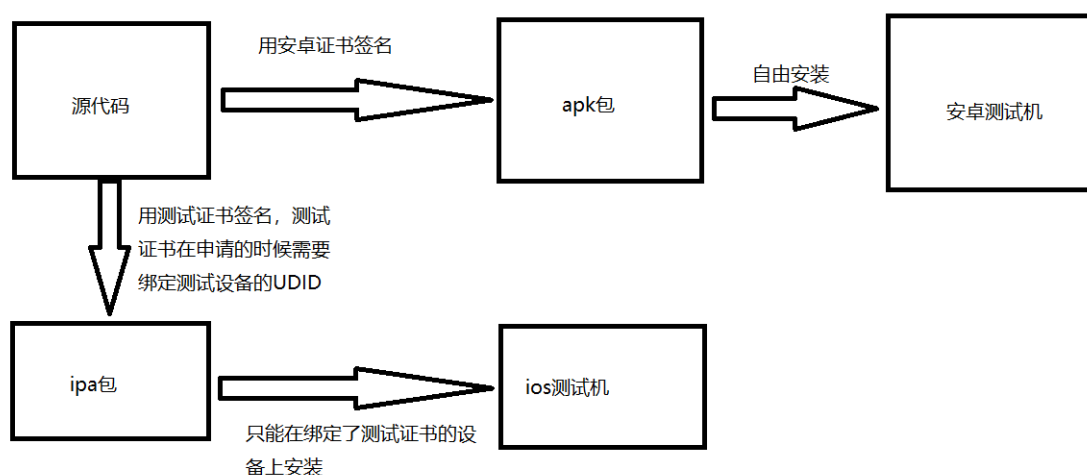
面试题：Web测试和App测试有什么区别？

功能测试方面，Web和App没有区别，App有一些专项的测试：安装、卸载、升级，兼容性（关注的点不一样），弱网，交叉事件，耗电量，流量，边界测试，压力测试（关注的点不一样）等。

7. uni-app打包

IDE工具：HBuilder X

打包前需要准备签名证书，Android证书的生成是自助和免费的，不需要审批或付费，iOS证书需要向苹果官方申请。其中iOS证书分为测试证书和正式证书，测试证书用于打测试版的App（可以在绑定了UDID的苹果设备上安装），正式证书用于打最终上架的App。



说明：用ios的证书签名，ios的证书需要申请，分为测试证书和正式证书，测试证书用来打测试的ipa包，正式证书用来打上架的ipa包。

App的包名：就是App的名字，包名一般用反写域名的方式来命名

一级域名：www.jd.com

二级域名：discount.jd.com ---->包名：com.jd.discount

8. Android模拟器的使用

- 模拟器的作用
 - 在实际工作中，因为某些原因，没有足够的测试机的时候，可以通过模拟器来模拟部分手机进行测试。
 - 但需要注意的是，模拟器始终跟真机存在差异，如果想确保App在真机上运行没有问题，必须在真机上进行测试。
- 常用的Android模拟器有：夜神、逍遥、MUMU、雷电等
- iOS模拟器：因为iOS是闭源的，做iOS模拟器的技术难度较大，目前国内的iOS模拟有黑雷、水果等，但都是收费软件。
- 演示：夜神模拟的使用。

9. 常用的adb命令

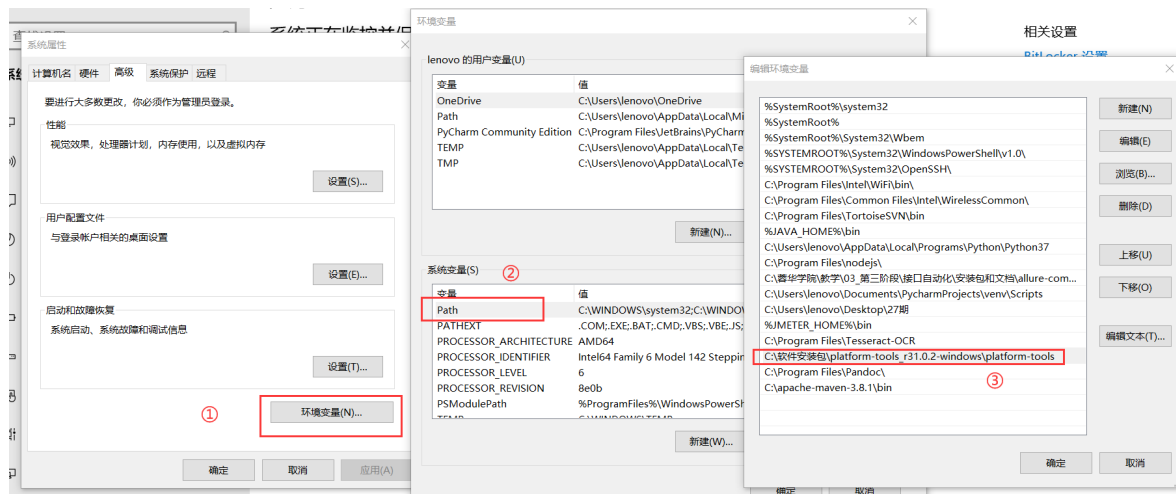
9.1 什么是adb——实现对Android设备的操作

adb: Android Debug Bridge, 安卓调试桥, 是一种命令行工具, 可实现对Android设备的操作, 比如安装、卸载apk, 往安卓设备上传、下载文件, 打印安卓设备的日志等。

9.2 adb环境配置

下载地址: <https://android-sdk.en.softonic.com/>, 下载最新版本的Android SDK, 解压。

配置环境变量, 把adb命令的目录配置到PATH中。记住: 配置Path环境变量的时候, 只能增加, 不能覆盖之前的配置。(配置环境变量的时候, 有三个页面的确定按钮需要点击)



检查是否配置成功, 在dos命令中输入adb version回车, 出现版本信息表示配置成功

```
C:\Users\lenovo>adb version
Android Debug Bridge version 1.0.41
version 29.0.2-5738569
Installed as C:\android-sdk-windows\android-sdk-windows\tools\adb.exe
```

通过adb连接Android设备 (是指真机), 有两种方法 (USB方式及网络方式):

方法1: 通过数据线连接, **注意是数据线不是充电线**, 在手机上先开启开发者模式, 然后再开启USB调试模式, 然后在dos窗口敲adb devices命令查看是否连接成功, 如果显示如下就表示连接成功 (要注意不是充电模式。连不上的下载手机助手)

```
C:\Users\lenovo>adb devices
List of devices attached
D5F0219815004432      device
```

步骤: 1、打开开发者模式; 2、开启USB调试。

方法2: 通过Wi-Fi连接, 首先将电脑和手机连接到同一个Wi-Fi (也要允许USB调试), 然后将手机通过数据线连接到电脑, 执行adb tcpip 5555在手机上开启adb无线模式服务 (这一句话的意思是 adb 从usb 模式转化成 adb 网络模式 (wifi))

```
C:\Users\lenovo>adb tcpip 5555 (在手机上开启adb无线模式服务并配置端口)
restarting in TCP mode port: 5555 (出现这句话代表端口号设置成功了 (意思是在tcp模式下重启端口号))
```

最后在dos窗口执行adb connect 手机IP连接手机, 如下就表示连接成功了

```
C:\Users\lenovo>adb connect 192.168.2.101:5555
connected to 192.168.2.101:5555

#断开Wi-Fi连接
C:\Users\lenovo>adb disconnect 192.168.2.101:5555
```

扩展: IP地址、掩码地址、网关地址的关系

网络连接详细信息(D):

属性	值
连接特定的 DNS 后缀	
描述	Intel(R) Wireless-AC 9560 160MHz
物理地址	98-2C-BC-BF-34-66
已启用 DHCP	是
IPv4 地址	192.168.7.82
IPv4 子网掩码	255.255.255.0
获得租约的时间	2021年7月29日 8:48:41
租约过期的时间	2021年7月30日 9:41:12
IPv4 默认网关	192.168.7.1
IPv4 DHCP 服务器	192.168.7.1
IPv4 DNS 服务器	183.221.253.100
	8.8.8.8
IPv4 WINS 服务器	
已启用 NetBIOS over Tcpip	是
连接-本地 IPv6 地址	fe80::9dc6:bb4d:3551:d894%8
IPv6 默认网关	
IPv6 DNS 服务器	

192.168.7.82 11000000.10101000.00000111.01010010

255.255.255.0 11111111.11111111.11111111.00000000 ----24个1，所以也叫24位掩码
192.168.7.82/24

192.168.7.0 11000000.10101000.00000111.00000000 -----网络地址 192.168.7.0/24

对IP地址192.168.7.82的解释是：网络地址是192.168.7.0，主机地址是82，主机地址的范围是0~255，其中0表示网络地址，255表示广播地址，1通常用来作为网关地址，所以在这个网络中可用的主机地址有2-254，总共253个。

$1 \& 1 = 1$

$1 \& 0 = 0$

$0 \& 1 = 0$

$0 \& 0 = 0$

9.3 常用的adb命令

- 查看adb的版本号：adb version

```
C:\Users\lenovo>adb version
Android Debug Bridge version 1.0.41
Version 29.0.2-5738569
Installed as C:\android-sdk-windows\android-sdk-windows\tools\adb.exe
```

- 查看连接设备及状态：adb devices，查出来的设备状态有3种
 - offline 表示离线（把USB调试关闭就可以看到）
 - device 表示连接正常
 - unauthorized 表示未授权（如果手机上没有允许USB调试，就会显示未授权）
- 开启Android设备的adb无线模式服务：adb tcpip 端口，比如adb tcpip 5555
- 通过Wi-Fi的方式连接设备：adb connect 设备IP:端口
- 断开设备：adb disconnect 设备IP:端口
- 安装app：
 - adb install apk的路径，如果是在真机上安装，需要在手机上点击弹出的允许安装才能成功
 - adb install -r apk的路径，-r表示覆盖安装，如果之前已经安装了该apk，直接安装会失败，需要覆盖安装

```
C:\Users\lenovo>adb install C:\蓉华学院\项目\占数家营销助手\安装包
\PromotionAssist-debug.apk
Performing Streamed Install
Success
```

- 卸载app：
 - adb uninstall 包名：卸载apk并删除数据
 - adb uninstall -k 包名：卸载apk，保留数据

```
C:\Users\lenovo>adb uninstall com.summer.evs
Success
```

- 查看所有包名：adb shell pm list package

包名是app的唯一标识，一般采用反写域名的方式命名，比如com.taobao.taobao

www.ronghuanet.com 一级域名

students.ronghuanet.com --->com.ronghuanet.students

teachers.ronghuanet.com --->com.ronghuanet.teachers

- 查看系统包名：adb shell pm list package -s
- 查看第三方包名：adb shell pm list package -3
- 查看包的安装位置：adb shell pm list package -f (系统包名或者第三方包名结合-f使用)
- 查找指定app的包名：adb shell dumpsys activity activities | findstr mResumedActivity (执行命令之前打开应用在前台运行)

```
C:\Users\lenovo>adb shell dumpsys activity activities | findstr
mResumedActivity
    mResumedActivity: ActivityRecord{84a6383 u0
com.ifeng.news2/.activity.IfengTabMainActivity t185}
```

- 查看app的版本号：adb shell pm dump 包名 | findstr "version"

举例：adb shell pm dump com.test.hm | findstr "version"

注：命令中的参数：com.test.hm是app的包名。

```
C:\Users\lenovo>adb shell pm dump com.sidoumi.his | findstr "version"
    versionCode=100 minSdk=19 targetSdk=28
    versionName=1.0.0
    signatures=PackageSignatures{a14bc9f version:2, signatures:[637809ff],
past signatures:[]}
```

- 打开APP：adb shell am start 包名/活动窗口名

```
C:\Users\lenovo>adb shell am start
com.ifeng.news2/.activity.IfengTabMainActivity t185
Starting: Intent { act=android.intent.action.MAIN cat=
[android.intent.category.LAUNCHER]
cmp=com.ifeng.news2/.activity.IfengTabMainActivity }
```

- 关闭打开的APP（放到后台）：adb shell am force-stop 包名

```
C:\Users\lenovo>adb shell am force-stop com.ifeng.news2
```

- **进入shell：adb shell，进入shell后可以执行linux命令**

- 如果电脑连接了多个设备，使用adb -s 设备名 shell来指定进入某个设备（比如：adb -s 127.0.0.1:62025 shell）

- **把Android设备上的文件下载到本地：adb pull Android设备上文件路径 本地目录的路径**（凡是以adb开头的命令都是在DOS窗口执行的）

- 用途：从测试设备上取文件、日志等到本地，给开发分析。

- **把本地文件上传到Android设备中：adb push 本地文件所在路径 Android设备的目标路径**（比如：adb push C:\Users\Administrator\Desktop\40qi\demo.txt /data/app，注意路径中不要有中文）

adb push: 把本地文件推到Android设备上, 相当于上传文件, 比如adb push C:\Users\Administrator\Desktop\123\app.log /data/anr 表示把windows上的 C:\Users\Administrator\Desktop\123\app.log上传到Android设备上的/data/anr目录 (如果没有写入权限: adb remount)

- 用途: 当开发修改完一个问题的时候, 如果要快速验证, 一般开发会提供一个补丁文件, 需要把这个文件推到Android设备的某个目录下替换原有的文件, 替换后重启App再次验证;
- 删除文件: adb shell rm 文件, 也可以先执行adb shell进入到安卓设备上, 然后通过rm命令删除
- **打印日志:**
 - **adb logcat** (要记住) 和 adb shell logcat 他们打印的内容是一模一样的, 区别在于adb logcat中的logcat是以参数的形式存在, 而adb shell logcat中的logcat是以shell 命令的形式存在 (会打印出历史的日志, 而我们关注的是与当前时间接近的那个时间)
 - **adb logcat: 打印日志到屏幕**
 - adb logcat *:W 日志筛选
 - adb logcat -v time *:w> c:\2111.txt 加时间 l < d < w < e
 - adb logcat -c: 清除缓存的日志, 在打印日志前可以先清理缓存的日志再打, 这样会更有针对性
 - adb logcat>C:\Users\Administrator\Desktop\123\app.log: 表示重向日志到电脑的C盘下的某个文件, 相当于保存日志文件到本地, 方便查看
 - **adb logcat -c && adb logcat>C:\Users\Administrator\Desktop\123\app.log: 先清空再输出**日志到指定位置 (>表示重定向)
 - 日志分析:
 - **测试结果分析: ANR (andriod系统一种自身监测机制), Exception, Null, Error, Crash**
- 1, ANR(程序无响应问题): 在日志中搜索“ANR”(applicationno response)
 - 2, 闪退问题: 在日志中搜索“crash”
- 3, 异常问题: 在日志中搜“Exception”基类
 - 4, 强制退出: 在日志中搜“forceclosed”
- 在Android上, 如果你的应用程序有一段时间响应不够灵敏(5秒内没有输入响应事件), 系统会向用户显示一个对话框, 这个对话框称作应用程序无响应 (ANR: ApplicationNot Responding) 对话框。
- 崩溃问题: 在日志中搜索“Exception”
- Null指针异常: 空指针异常主要是有 NullPointerException 异常提示
- debug异常: 出现的是 IllegalStateException 异常
- 低内存异常: 出现的是OutOfMemoryError 异常
- 操作无响应异常: TimerOut (KeyDispatchingTimedOut提示)
- RuntimeException异常 (操作无响应或应用的服务无法启动或连接)
- StaleDataException/readException异常 (android 数据库出现异常)
- IllegalArgumentException异常 (向函数传递了一个不正确或不合法的参数)

ANR: Application Not Responding, 应用无响应, 当一个app出现一次ANR问题时, Android系统会自动生成一个日志文件放到/data/anr目录下, 这时需要把这个文件取下来分析。

##10. monkey压力测试 (稳定性测试——通过随机点击屏幕一段时间, 看看app会不会崩溃, 能不能维持正常运行)

- 什么是monkey

- adb monkey是Android系统一款稳定性测试工具，通过随机操作app一段时间，看app会不会出现崩溃、无响应，是否能够正常维持运行的测试。是由android系统自带应用，可以通过adb shell monkey直接使用。（伪随机）
- monkey命令的路径（monkey工具所在的路径）
-

```
127|HWCLT:/ $ ls -l /system/bin/monkey
-rwxr-xr-x 1 root shell 268 2018-08-08 00:01 /system/bin/monkey
```

- monkey的12种随机事件：（支持12种随机操作，每一种有一个编号）

- 编号0：触摸事件，参数--pct-touch，触摸事件指的是在屏幕某处按下并抬起的操作，我们可以模拟点击的操作；
- 编号1：手势事件（滑动事件），参数--pct-motion，滑动事件指的是在屏幕某处按下、随机移动、抬起的操作，类似我们滑动手机屏幕翻页的操作。
- 编号2：二指缩放事件，参数--pct-pinchzoom，缩放事件指的是在屏幕上的两处同时按下，又同时移动，最后同时抬起，模拟我们在手机放大缩小的操作
- 编号3：轨迹事件，参数--pct-trackball，指的是由一个或者多个随机的移动组成，有时会伴随点击。
- 编号4：屏幕旋转事件，参数--pct-rotation，模拟手机的横竖屏切换，degree指的是旋转方向，顺时针旋转，0表示旋转90°，1表示旋转180°，2表示旋转270°，3表示旋转360°
- 编号5：permission事件
- 编号6：基本导航事件，参数--pct-nav，指的是点击方向输入设备的上下左右按键的操作，现在的手机很少有这样的方向键了，所以应用较少
- 编号7：主要导航事件，参数--pct-majornav，指的是点击“主要导航”按键的操作，例如app的中间键，返回键，菜单按键等
- 编号8：系统按键事件，参数--pct-syskeys，指的是系统保留的按键事件，比如Home键、返回键、音量调节按键等
- 编号9：启动activity事件，参数--pct-appswitch，指的是在手机上启动一个Activity的操作，在随机的时间间隔内，Monkey 将执行一个 startActivity() 方法，作为最大程度上覆盖被测包中全部 Activity 的一种方法
- 编号10：键盘事件，参数--pct-flip，键盘事件指的是与键盘相关的操作，比如点击输入框、键盘弹起等
- 编号11：其他类事件，参数--pct-anyevent，指的是除了上述事件之外的其他事件，比如其它不常用的设备按钮等

- 举例：adb shell monkey 1000 表示对手机整机进行1000次随机操作

- 部分参数：

- -p：指定有效的package（如不指定，则对系统中所有package有效），一个-p对应一个有效package，如：-p com.ckt -p com.ckt.asura；
- -v：指定打印信息的详细级别，一个-v增加一个级别，默认级别为0，三个-v表示最详细的级别。
- -s：指定产生随机事件种子值（seed），相同的种子值产生相同的事件序列。如：-s 200（这种用法可以让问题重现

1) 用于指定伪随机数生成器的seed值，如果seed相同，则两次monkey测试所产生的事件序列也相同

命令：adb shell monkey -p com.baidu.searchbox -v -s 88 10提前指定88为seed值

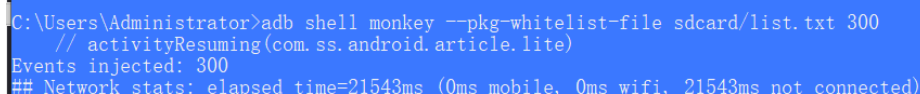
再次执行：adb shell monkey -p com.baidu.searchbox -v -s 88 100

说明：两次执行的效果是相同的，因为模拟的用户操作序列（每次操作按照一定的先后顺序所组成的一系列操作就是一个序列）是一样的，操作序列虽然是随机生成的，但是只要我们制定了相同的seed值，就可以保证两次测试产生的随机操作序列是完全相同的，所以这个操作序列是伪随机的)

- --throttle：每个事件结束后的间隔时间——降低系统的压力（如不指定，系统会尽快的发送事件序列）。如：--throttle 100
- 指定各种事件百分比的参数：
 - --pct-touch：指定触摸事件的百分比，如：--pct-touch 5 表示触摸事件占5%的比例
 - --pct-motion 百分比（滑动事件）
 - --pct-pinchzoom 百分比（二指缩放事件）
 - --pct-trackball 百分比（轨迹球事件）
 - --pct-rotation 百分比（屏幕旋转事件）
 - --pct-nav 百分比（导航事件 up/down/left/right）
 - --pct-majornav 百分比（主要导航事件 back key、menu key）
 - --pct-syskeys 百分比（系统按键事件 Home、Back、startCall、endCall、volumeControl）、--pct-appswitch 百分比（activity之间的切换）
 - --pct-appswitch 百分比（启动activity事件）
 - --pct-flip 百分比（键盘时间）
 - --pct-anyevent 百分比（任意事件）
- 调试选项：
 - --dbg-no-events：初始化启动的activity，但是不产生任何事件。
 - --hprof：指定该项后在事件序列发送前后会立即生成分析报告——一般建议指定该项。
 - **--ignore-crashes：当应用程序崩溃或发生任何失控异常时，monkey将停止运行。如果设置此选项，monkey将继续向系统发送事件，直到计数完成。**
 - **--ignore-native-crashes：忽略本地代码导致的崩溃。设置忽略后，Monkey将执行完所有的事件，不会因此停止**
 - **--ignore-timeouts：通常，当应用程序发生任何超时错误（如'Application Not Responding'对话框）时，monkey将停止运行。如果设置了此选项，monkey将继续向系统发送事件，直到计数完成。**
 - **--ignore-security-exceptions：忽略安全异常**
 - --kill-process-after-error：发生错误后直接杀掉进程
 - --monitor-native-crashes：跟踪本地方法的崩溃问题
 - --wait-dbg：知道连接了调试器才执行monkey测试。
 - 一个比较完整的monkey命令（注意，要将monkey信息输出到文件中，随机事件的操作次数要写在重定向前面）
 - 一个简单的monkey命令：adb shell monkey -p com.xy.android.junit -s 1635966164169 -v -v -v 10000
 - 但是，工作中为了保证测试数量的完整进行，我们一般不会在发生错误时立刻退出压力测试。monkey测试命令如下：
 - adb shell monkey -p com.xy.android.junit -s 1635966164169 --ignore-crashes --ignore-native-crashes --ignore-timeouts --monitor-native-crashes -v -v -v 10000 >E:\monkey_log\java_monkey_log.txt 2>&1
 - 其中，2>&1表示将输出和错误都重定向到文件中
 - --pkg-blacklist-file：黑名单，排除某些包不执行，如--pkg-blacklist-file /data/blacklist.txt

- --pkg-whitelist-file: 白名单, 指定执行哪些包, 如--pkg-whitelist-file /data/whitelist.txt

- 如图



```
C:\Users\Administrator>adb shell monkey --pkg-whitelist-file sdcard/list.txt 300
// activityResuming(com.ss.android.article.lite)
Events injected: 300
## Network stats: elapsed time=21543ms (0ms mobile, 0ms wifi, 21543ms not connected)
```

- 使用举例:

- **命令一: 用monkey去跑指定的app跑1万次, 忽略异常和超时, 把输出和错误都重定向到文件中**

```
adb shell monkey -p tv.danmaku.bili -v -v -v --ignore-crashes --ignore-native-crashes --
ignore-timeouts --ignore-security-exceptions 10000>C:\Users\lenovo\Desktop\37期
\monkey7.txt 2>&1
```

- **命令二: 指定seed值用monkey去跑指定的app跑1万次, 忽略异常和超时, 把输出和错误都重定向到文件中, 这个用来复现或回归问题**

```
adb shell monkey -p tv.danmaku.bili -v -v -v --ignore-crashes --ignore-native-crashes --
ignore-timeouts --ignore-security-exceptions -s 1637762067514
10000>C:\Users\lenovo\Desktop\37期\monkey7.txt 2>&1
```

- 如果你想跑到出问题就停止, 那就不需要忽略异常, 把跑的次数设置足够大。

在重定向的日志中搜索**crash**、**anr**、**error**、**exception**等关键字看有没有出现过类似错误。

常见报错: 部分小米手机在执行adb shell monkey命令时报java.lang.SecurityException: Injecting to another application requires INJECT_EVENTS permission, 解决方法是在开发者选项页面, 打开USB调试(安全设置), 允许通过USB调试修改权限或模拟点击。

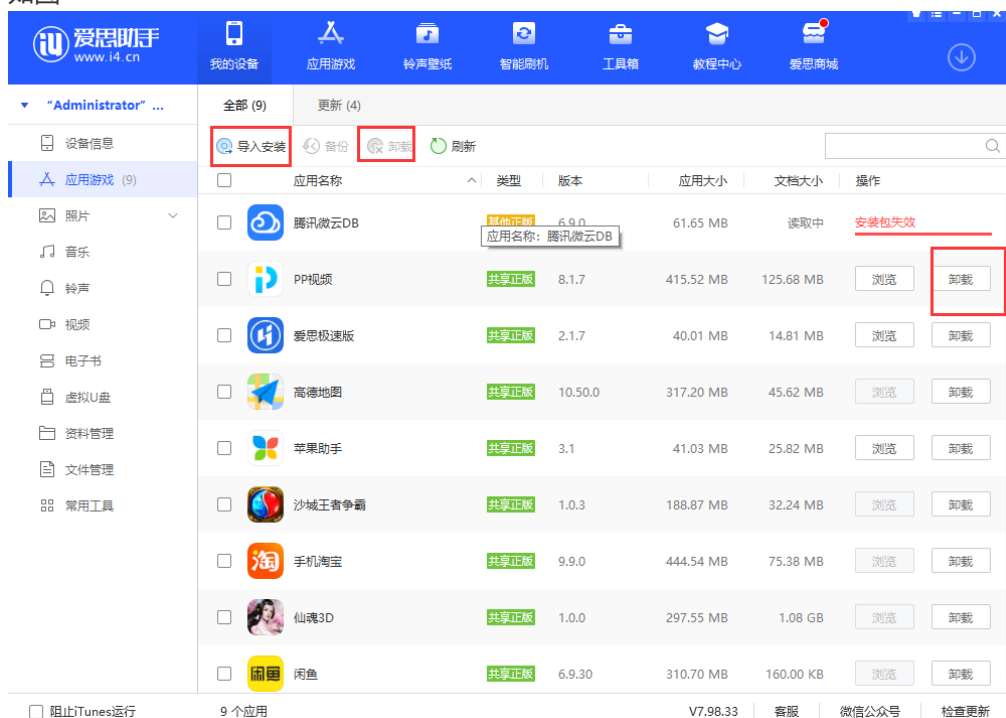


11. 助手工具的使用

- 91助手（都支持）、豌豆荚（Android）、爱思助手（iOS）、PP助手（都支持）、360手机助手（都支持）等等；
- 在测试中，可以借助这些助手工具完成测试App的安装、卸载，升级测试等，包括iOSApp和Android App；
- 以91助手为例，要通过助手管理手机，需要：
 - 1) 开启手机的开发者模式，再打开USB调试；
 - android手机需要做这个操作，步骤：打开设置-->打开关于手机-->找到版本号连续点击3次以上，直到看到进入开发者模式-->找到开发者模式并进入--->打开usb调试-->然后用数据线连接电脑，在配对信息中点允许配对
 - 2) 用数据线连接PC与手机，按提示点击确定，91助手会在手机上安装一个91助手App，允许其安装即可；
 - 3) 正常情况下就可以连接PC与手机了，把测试App拖到应用一栏里，就可以安装App；
 - 注意：
 - 1) 测试的iOS App安装，需要提前获取到测试设备UDID，把UDID写到测试App中一起打包，这样才能在测试设备上安装这个App；并且信任
 - 2) 测试App中使用的是测试证书，在安装完成测试iOSApp后，还需要在设置下面信任测试证书，才能使用这个测试App；

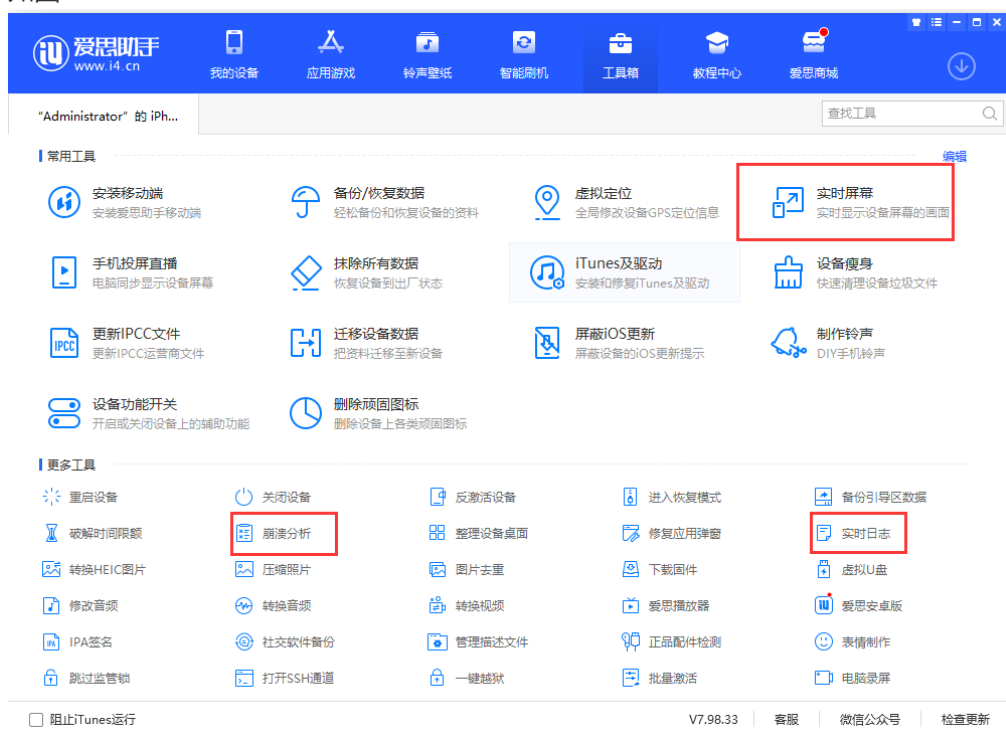
- 3) 91助手可以通过iOS设备日志查看苹果设备的日志信息。
- 安装卸载升级操作：

■ 如图



- 截图，日志，崩溃分析

■ 如图



使用助手工具的主要场景：安装、卸载app，打印iOS的日志等。

12. 弱网测试工具

造成弱网的原因有：**网络延迟**（由于网络太差，产生了网络波动，使数据包在传输的时候出现抖动——抖动是延迟的一种，是指网络发生拥塞的情况下，排队产生的延迟会影响端到端的延迟，并导致通过同一连接传输的分组延迟各不相同，而抖动就是用来描述这样一延迟变化的程度，一般网络抖动值指的是网络通信中延迟最大值与最小值之差，网络抖动值越小说明网络质量越稳定，比如在地铁中）、**丢包**

(是指一个或者多个数据包无法通过网络来达成目的。换句话说网络在传输的过程中,数据是以包的形式进行传输的,在TCP协议中,需要不停的发送请求,来确认连接,一旦发生丢包,就需要重传;如果app在运行的过程中,出现了异常,导致数据丢包了,我们要检测下产品的处理机制是否会出来?如果程序未响应,程序如何处理,会给与什么提示信息,比如看视频不正常,浏览网页,如果网络丢包,会感觉很慢,因为丢包要等待重传。丢包率,小于5%,网络可以用;5% - 10%之间,有明显的慢的感觉,高于10%,很慢,高于30%,网络基本不可用)、**带宽**(用来表示网络的通信线路所能传送数据的能力,因此网络带宽表示在单位时间内从网络中的某一点到另一点所能通过的“最高数据率”)

总结:丢包:就是你传送数据时,有没有掉失,掉包多了,证明网络质量差。网络延迟:就是你发数据到对方要多久时间,网络延迟高了,就是俗称的"卡"

生活中的典型弱网场景:电梯、地下室、地铁(高峰、错峰)、公交(高峰、错峰)、高铁(速度快,频繁切换基站)、山里

弱网测试的基本标准:app在弱网情况下不出现闪退、崩溃、ANR等问题。

弱网测试工具

1. **当前推荐工具: QNET** (腾讯wetest 服务平台推出的一款针对 App 弱网测试工具,该工具无需 ROOT 手机,无需连接数据线,以独立 app 的方式,为用户提供给快捷、可靠、功能完善的弱网络模拟服务)
 - QNET是当前最好用的弱网测试工具,能够模拟延迟、丢包、抖动等弱网场景,它以独立的 app方式运行,操作简单,它里面内置了国内所有省份和海外部分国家的网络数据以及常见的弱网场景(公交高晚高峰、地铁早晚高峰、停车场、电梯等),在使用的时候直接选择内置的场景就可以了。
 - 安装后默认有连续丢包、正常网络、100%丢包、极差网络四个模板。



- 除此之外，还可以新增模板，新增模板时可以选择类型为场景模板，就可以直接选择QNET中预置的场景。

未命名

保存

模板名称

未命名

模板类型

场景模板 >

场景模板/自定义模板

网络选择

服务端

上海 >

选择服务端所在地

客户端

北京市 >

选择客户端所在地

运营商

中国移动 >

选择网络运营商

网络类型

4G >

WiFi或者蜂窝网络

取消

确定

正常网络

电梯(WiFi切流量)

电梯(信号极差)

电梯(信号良好)

高铁(基站切换)

2. 其它工具：360 WiFi、Fiddler（抓包工具）、切换手机网络制式

- 360 WiFi限速：

- A. 360免费WiFi



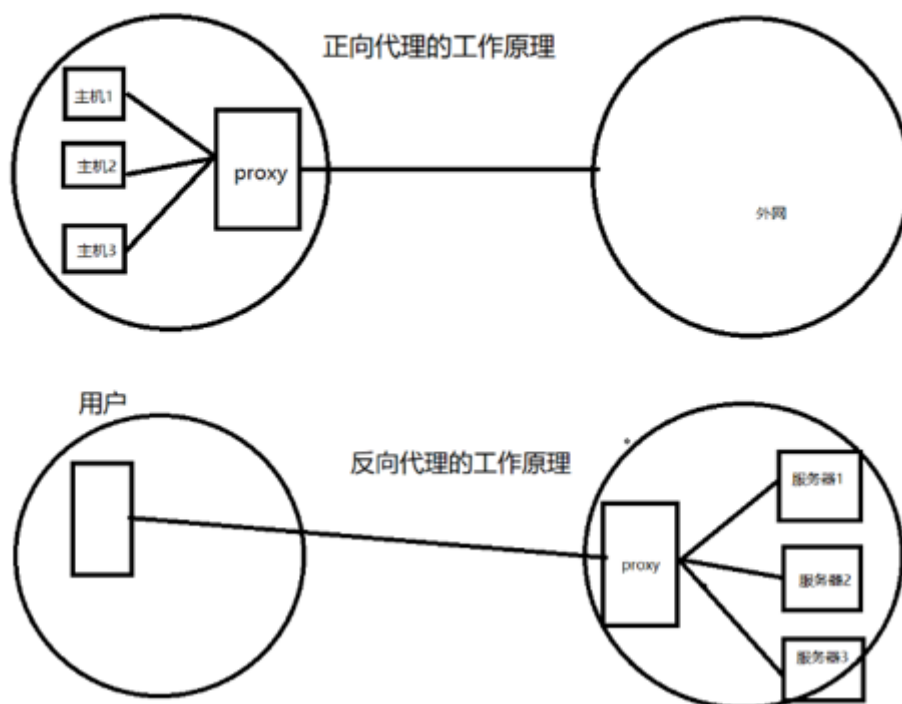
- 如果使用移动网络，通过在手机上选择不同的网络制式；

如图



- 通过fiddler限速
fiddler的原理

正向代理与反向代理的工作原理：



通过fiddler来实现限速步骤：

- (1) 在Rules-》CustomizeRules下面，首先要保证m_SimulateModem值为true，表示开启限速

```
// Cause Fiddler to delay HTTP traffic to simulate typical 56k modem conditions
public static RulesOption("Simulate &Modem Speeds", "Performance")
var m_SimulateModem: boolean = true;
```


(2) 如下两行代码分别表示对上传和下载的限制，第一行表示上传1KB的延时是300ms，相当于上传速度是1KB/300ms=3.33333KB/s，第二行表示下载1KB延时150ms，通过设置这两个延时来控制网速，延时越大网速越慢。

```
if (m_SimulateModem) {  
    // Delay sends by 300ms per KB uploaded.  
    oSession["request-trickle-delay"] = "3000";  
    // Delay receives by 150ms per KB downloaded.  
    oSession["response-trickle-delay"] = "1500";  
}
```

13. App的性能测试

13.1 什么是性能狗PerfDog

performance

PerfDog是腾讯WeTest团队研发的一款性能测试工具，是移动全平台iOS/Android性能测试、分析工具平台。快速定位分析性能问题，提升APP应用及游戏性能和品质。手机无需ROOT/越狱，手机硬件、游戏及应用APP也无需做任何修改，极简即插即用。

PerfDog支持移动平台所有应用程序（游戏、APP应用、浏览器、小程序、小游戏、H5、后台系统进程等）、Android模拟器、云真机等性能测试。支持APP多进程测试(如Android多子进程及iOS扩展进程APP Extension)。

13.2 性能狗的使用演示

参考：https://perfdog.qq.com/article_detail?id=10089&issue_id=0&plat_id=1

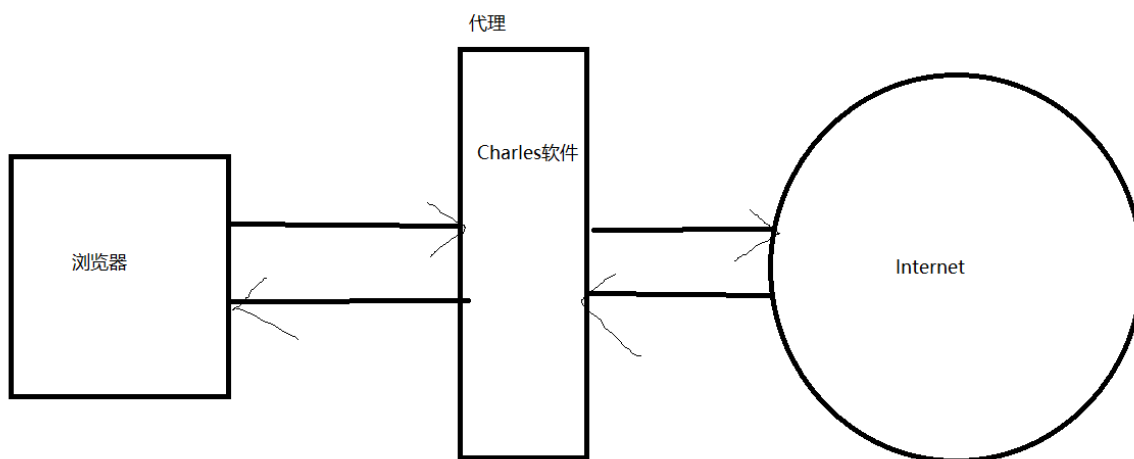
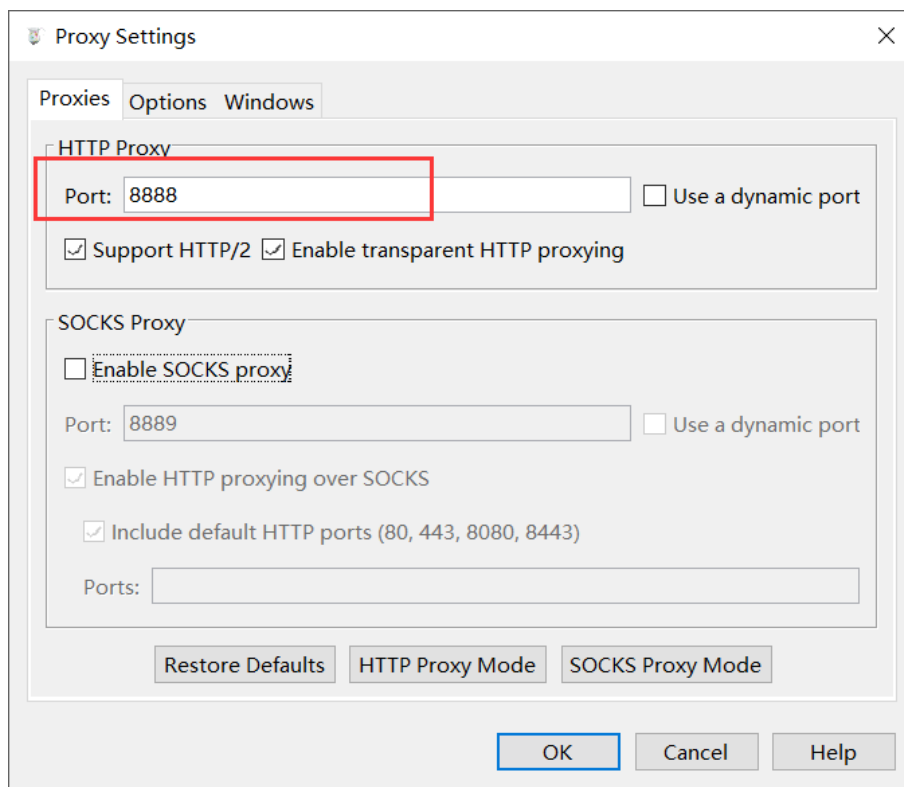


14. App抓包

主流的抓包工具：开发者工具F12、Charles、Fiddler、Wireshark等

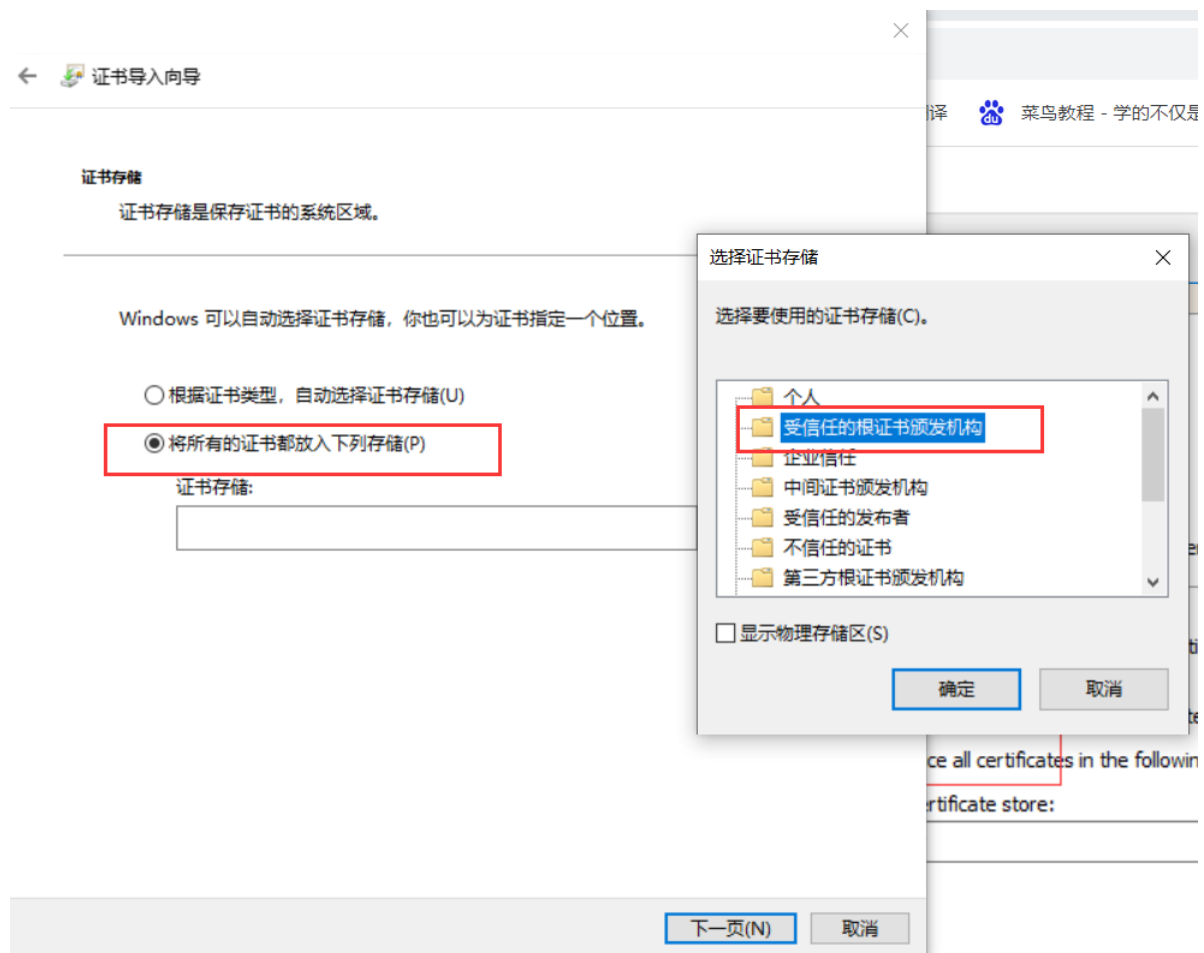
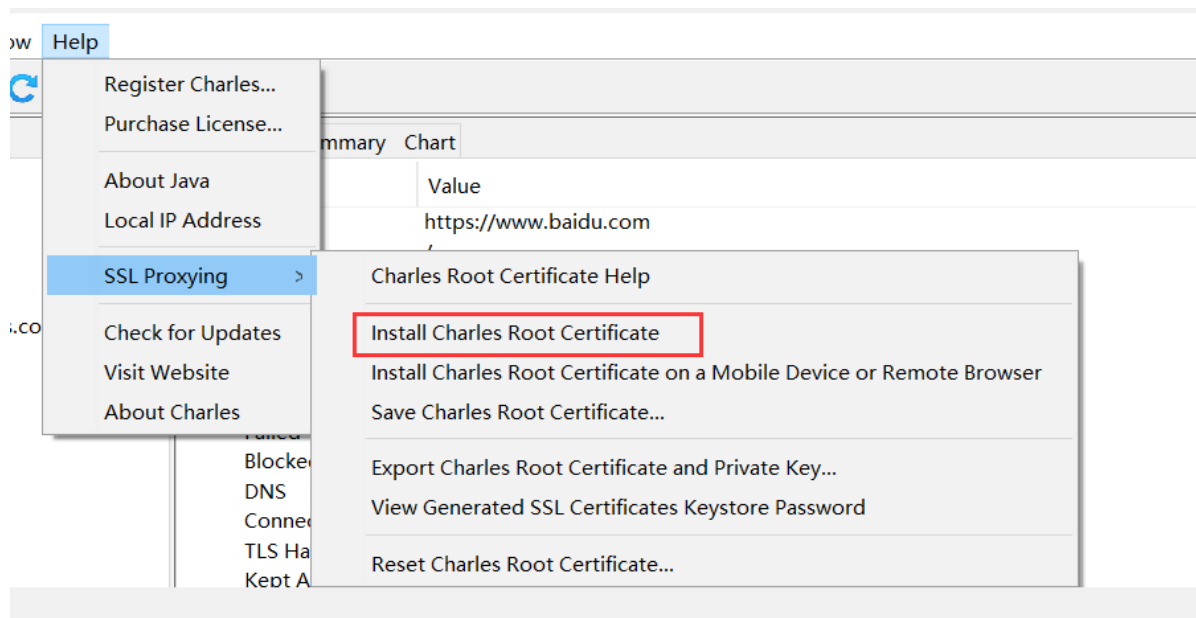
Charles抓包工具的使用

Charles/Fiddler抓包的原理：启动后会监听在8888端口，同时它会修改电脑的代理为http=127.0.0.1:8888;https=127.0.0.1:8888，表示Charles/Fiddler是作为电脑的代理工作。



https是加密的，http是不加密的，https=http+ssl

抓https的包，需要安装证书：Help->SSL Proxying->Install Charles Root Certificate，安装之后通过抓一个https的包来验证是否安装成功。



抓App的包的步骤：

- 1、把手机和PC连接到同一个Wi-Fi；
- 2、修改手机的代理，配置成PC的IP（Wi-Fi分配的IP）和端口（Charles的端口），保存；

(未更改) 

☒ 显示高级选项

代理 手动 >

该浏览器使用 HTTP 代理，但其他应用可能不会使用

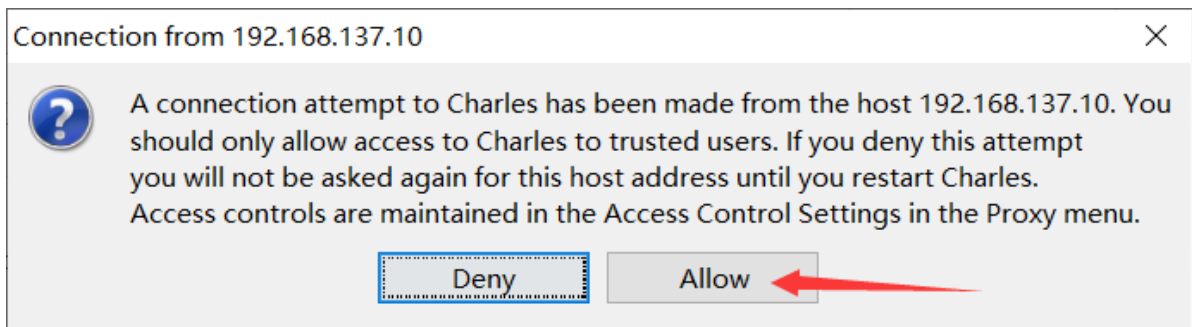
服务器主机名
192.168.2.102

服务器端口
8888

对以下对象绕过代理：
example.com,mycomp.test.com,localho:

IP DHCP >

3、在Charles上会弹出如下窗口，点击Allow运行手机连接Charles，到这一步，在Charles上就可以抓到手机的http包了，如果要抓https的包，继续第4步操作；

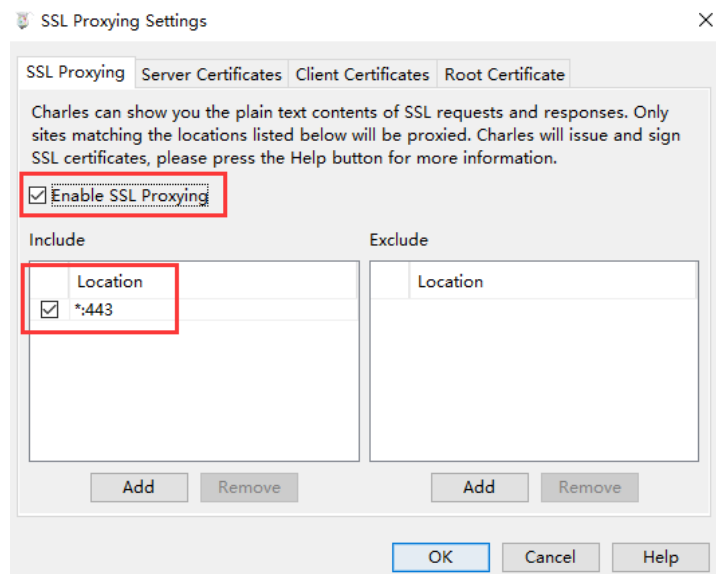


4、在Charles上开启SSL代理

Proxy -> SSL Proxying Settings-->勾选Enable SSL Proxying

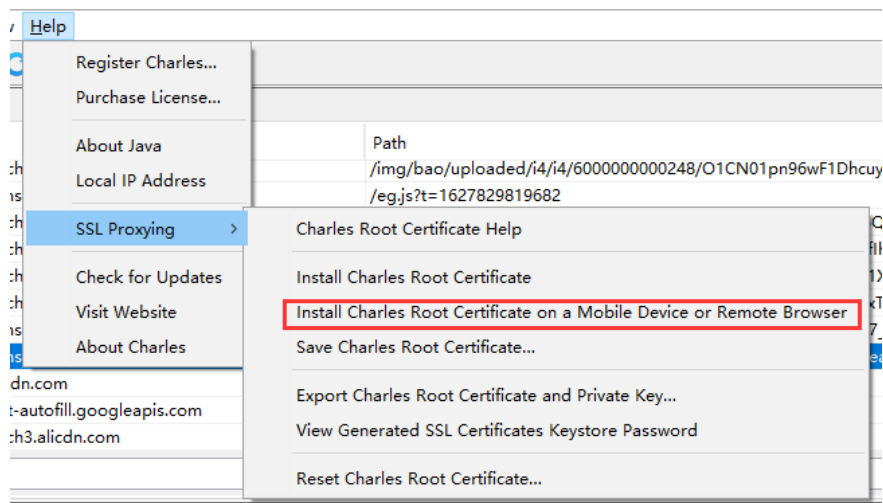
点击Add,点击Add, Host设置要抓取的https接口地址和端口

Host : * (使用通配符表示检测所有网络请求；建议还是设置单个需要抓取的https host，尽量避免使用 * 通配符)

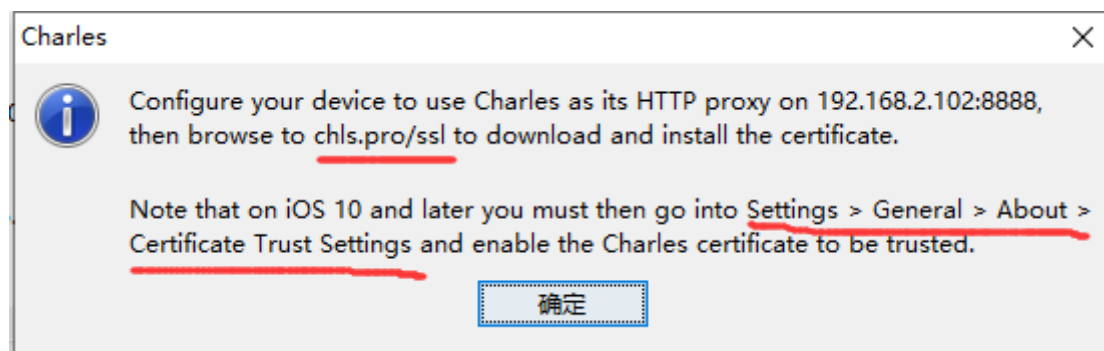


5、在手机端安装Charles证书

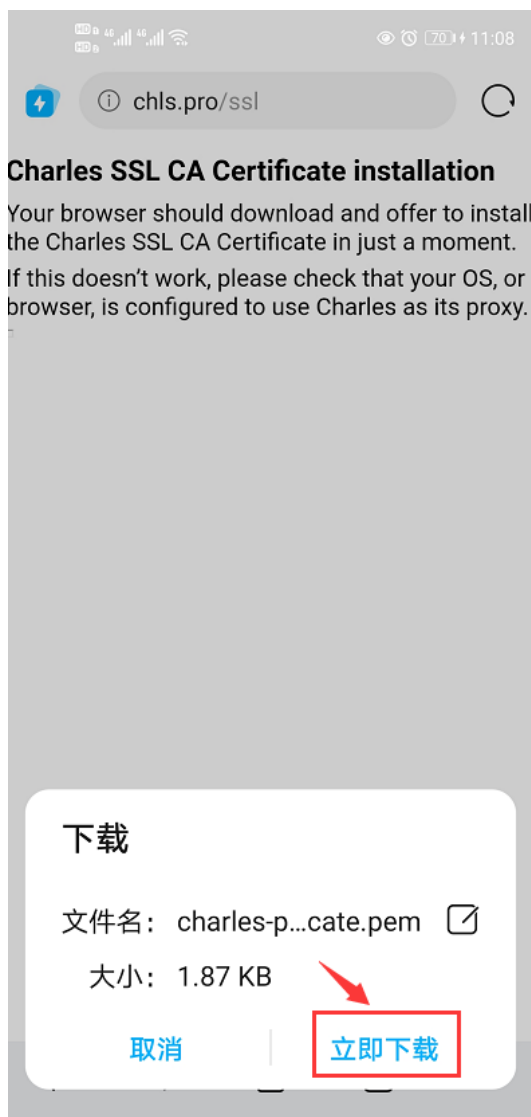
在Charles上点击Help->SSL Proxying --> Install Charles Root Certificate on a Mobile Device or Remote Browser,会提示你如何在手机端配置代理以及安装证书:



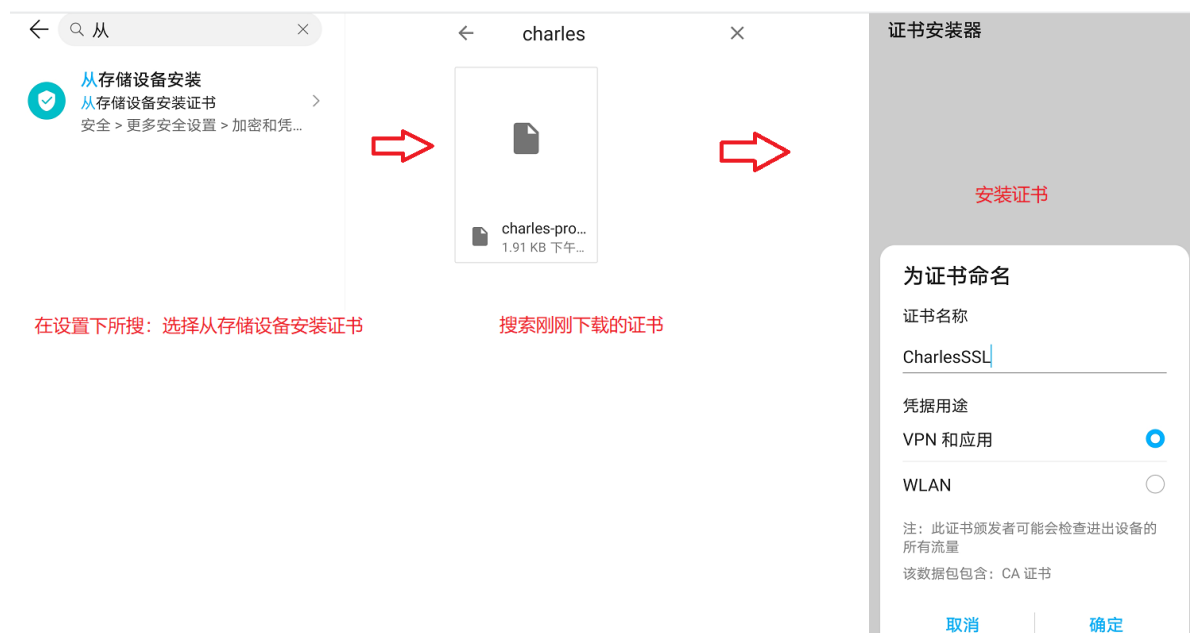
点击后提示如下:



然后在手机浏览器上打开<http://chls.pro/ssl>下载证书，下载到一个你能找到的目录，别下载完就找不到了。



下载后在手机的系统设置-》更多设置-》系统安全-》加密与凭据-》从存储设备安装，安装刚刚下载的证书。



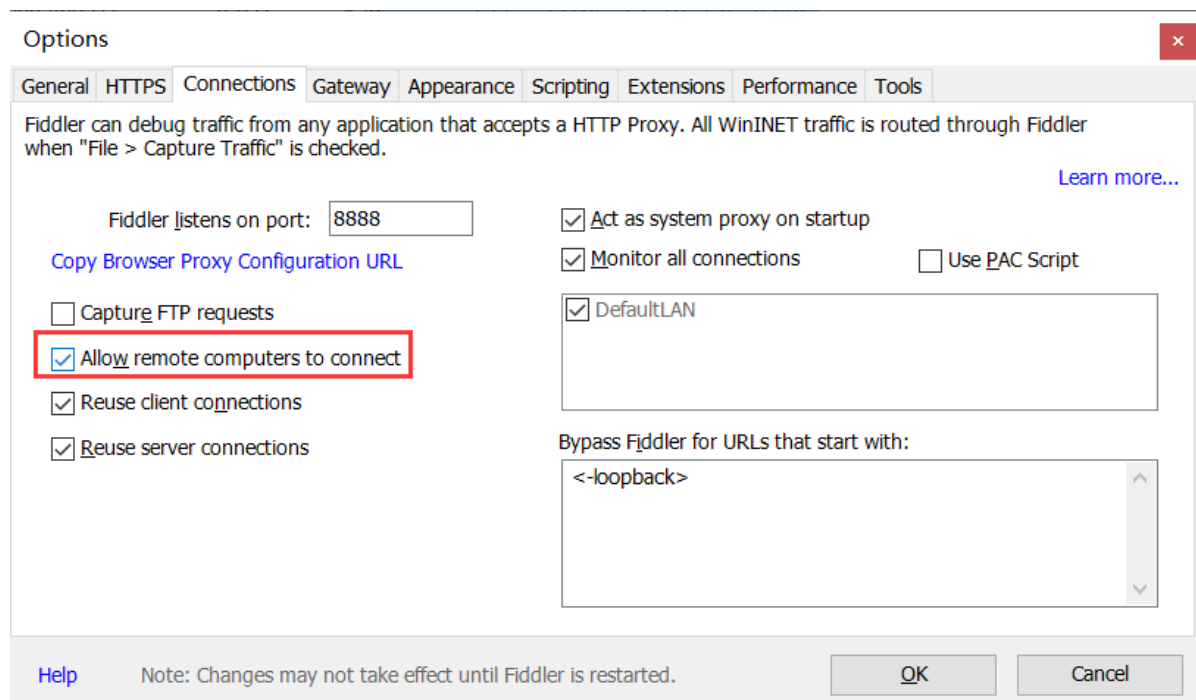
安装证书后，就可以抓到https的包了。

HTTP请求的常用方法：

- GET：表示从服务器获取资源
- POST：表示向服务器提交资源
- PUT：表示修改资源
- DELETE：表示删除资源

用Fiddler抓取手机App的包的步骤：

1、安装Fiddler，配置Fiddler，勾选允许远程电脑连接



2、把手机和电脑连接到同一个Wi-Fi里；

3、查到电脑的Wi-Fi的IP地址

属性

SSID:	RHStudent_2
协议:	Wi-Fi 5 (802.11ac)
安全类型:	WPA2-个人
网络频带:	5 GHz
网络通道:	149
链接速度(接收/传输):	650/866 (Mbps)
本地链接 IPv6 地址:	fe80::9dc6:bb4d:3551:d894%8
IPv4 地址:	192.168.100.158
IPv4 DNS 服务器:	192.168.100.1 192.168.100.1
制造商:	Intel Corporation
描述:	Intel(R) Wireless-AC 9560 160MH
驱动程序版本:	22.40.0.7
物理地址(MAC):	98-2C-BC-BF-34-66

4、连接Fiddler下载证书并安装

在手机浏览器中打开 <http://电脑IP:8888>

5、修改手机的代理，IP为电脑的Wi-Fi IP，端口为8888

6、在手机上打开浏览器，输入<https://m.baidu.com>，正常在Fiddler就可以抓到手机浏览器的包