

# 移动端测试基础

---

## 1 移动端自动化环境搭建

---

### 1.1 java安装

java JDK

安装 jdk-8u181-windows-x64.exe

配置环境变量：

JAVA\_HOME: D:\developer tools\Java\jdk1.8.0\_181

配置path: %JAVA\_HOME%\bin

%JAVA\_HOME%\jre\bin

验证：终端输入：java -version

```
C:\Users\ThinkPad>java -version
java version "1.8.0_181"
Java(TM) SE Runtime Environment (build 1.8.0_181-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)
```

### 1.2 Android SDK安装

解压Android-sdk.zip

配置环境变量

ANDROID\_HOME: D:\android-sdk-windows

配置path:

%ANDROID\_HOME%\platform-tools

%ANDROID\_HOME%\tools

验证

终端输入:

```
C:\Users\ThinkPad>adb version
Android Debug Bridge version 1.0.40
Version By Francesco Tescari
Installed as D:\android-sdk-windows\platform-tools\adb.exe
```

# 安装手机模拟器

推荐夜神模拟器

启动问题

夜神模拟器 bin目录下 nox\_adb.exe 需要和SDK中 adb.exe 版本一致

将SDK platform-tools下的adb.exe 拷贝到夜神bin目录更改名字--nox\_adb.exe 替换即可

启动方式

启动夜神模拟器，终端输入adb connect 127.0.0.1:62001

验证方式

```
C:\Users\ThinkPad>adb connect 127.0.0.1:62001
* daemon not running; starting now at tcp:5037
* daemon started successfully
connected to 127.0.0.1:62001

C:\Users\ThinkPad>
```

## 2 ADB Android调试桥

### 2.1 adb概念

adb概念

ADB 全名 Android Debug Bridge，是一个调试工具

adb是一个C/S架构的命令行工具，由三部分组成，ADB就是连接手机和电脑通信的桥梁，只要这个桥梁建立起来了，那么你就可以通过电脑来控制你的手机了

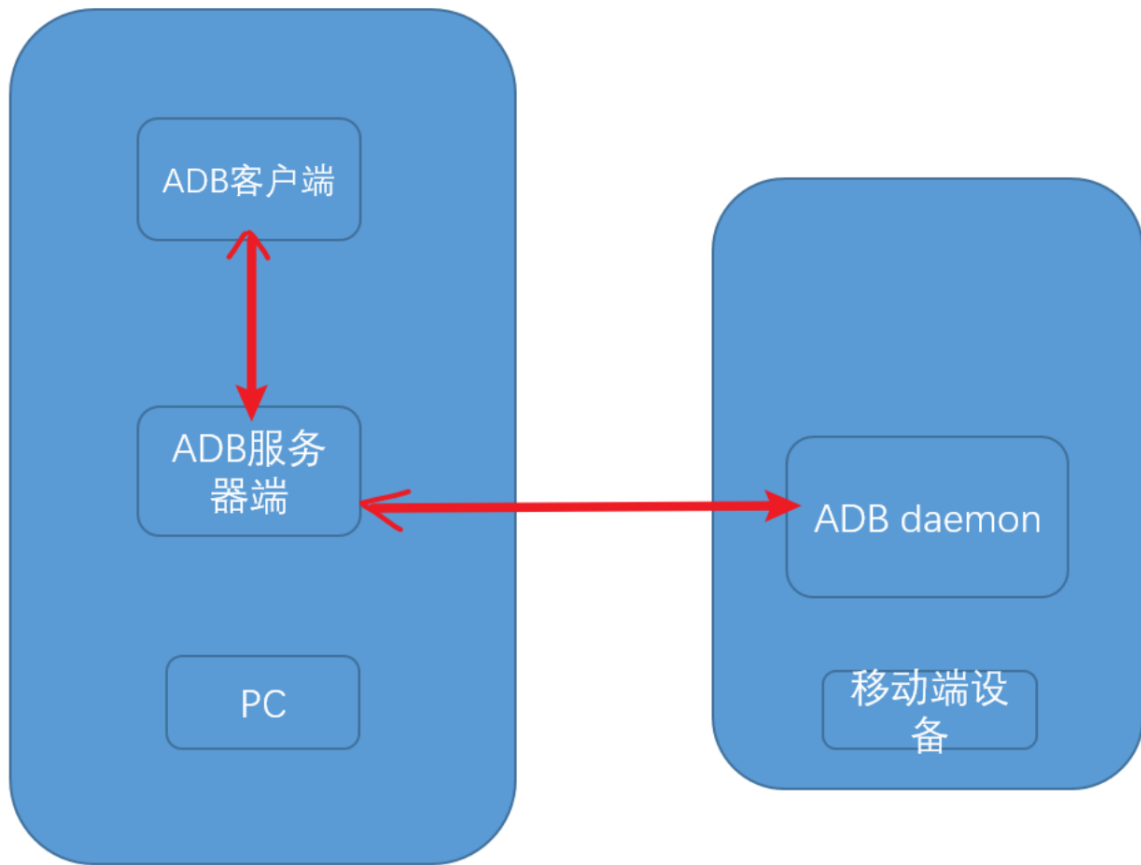
### 2.2 adb工作原理

adb 包含三个部分：

**Client**端：运行在pc端，用户操作界面，可以通过adb命令对 Android 应用进行安装、卸载及调试等；

**Daemon**：守护进程，运行在调试设备中，手机或模拟器，用来接收并执行 adb 命令；

**Server**端：运行在pc端中，客户端到 Android 设备上 adb 后台进程的连接，负责管理client和daemon进行通信。



## 2.3 adb常见命令

包名 (package) : 每个app的package是唯一的

界面名 (activity) : web页面中的链接地址

### 2.3.1 获取包名和界面名称命令

应用场景: 自动化测试中需要知道app的包名和activity, 通过命令获取

使用步骤: 开启被测应用, 输入命令

命令格式:

```
adb shell dumpsys activity|findstr Focused
```

在 Android 系统中, dumpsys 是一个用于获取系统服务信息的命令。

示例: 查询拉勾教育app

```
C:\Users\ThinkPad>adb shell dumpsys activity|findstr Focused
mFocusedActivity: ActivityRecord{14610856 u0 com.lagou.education/.ui.HelloActivity t2} 界面名
mFocusedStack=ActivityStack{2d39964d stackId=1, 1 tasks} mLastFocusedStack=ActivityStack{2d39964d stackId=1, 1 tasks}
```

### 2.3.2 adb文件传输命令

应用场景：发送文件到手机、从手机中拉取文件

**发送文件到手机命令格式：**

adb push 电脑的文件路径 手机的文件夹路径

示例：d盘下的apk 发送到 手机的sdcard目录下

```
C:\Users\ThinkPad>adb push d:/jisuanqi.apk sdcard
d:/jisuanqi.apk: 1 file pushed. 1.8 MB/s (1517570 bytes in 0.797s)
```

PC端目录 → 移动端文件夹

**手机中拉取文件命令行格式：**

adb pull 手机端路径 pc端文件夹

```
C:\Users\ThinkPad>adb pull /sdcard/screen.png d:/apk
/sdcard/screen.png: 1 file pulled. 3.0 MB/s (696154 bytes in 0.222s)
```

手机端路径 → PC端目录

### 2.3.3 获取app启动时间

应用场景：需要测试应用的启动速度

命令格式：

```
adb shell am start -w 包名/启动名
```

```
C:\Users\ThinkPad>adb shell am start -w com.lagou.education/.ui.HelloActivity
Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.lagou.education/.ui.HelloActivity }
Status: ok
Activity: com.lagou.education/.ui.HelloActivity
ThisTime: 625
TotalTime: 625
WaitTime: 647
Complete
```

ThisTime：该界面（activity）启动耗时（毫秒）

TotalTime：应用自身启动耗时 = ThisTime + 应用 application 等资源启动时间（毫秒）

WaitTime：系统启动应用耗时 = TotalTime + 系统资源启动时间（毫秒）

### 2.3.4 获取手机日志命令

应用场景:bug日志信息发给开发，方便开发定位问题

使用步骤：

开启应用 -- 执行bug产生位置 -- 使用命令 -- 获取日志信息

命令格式

```
adb logcat
```

常见命令

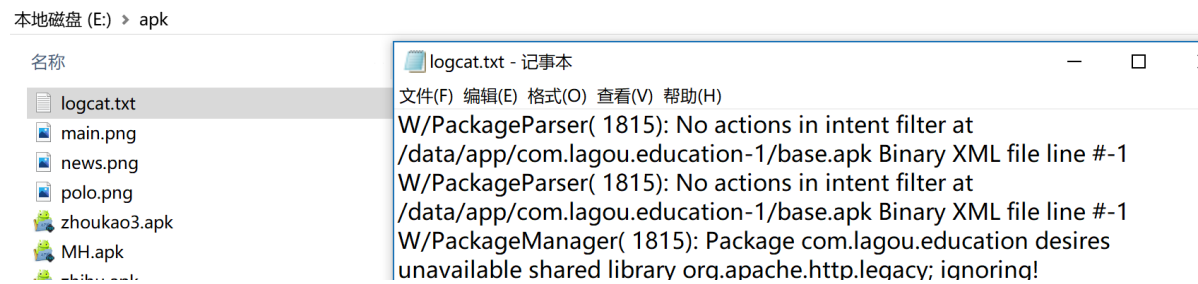
```
adb logcat|findstr com.flutter_luckin_coffee > E:/apk/logcat.txt
```

第一步：

```
C:\Users\ThinkPad>adb logcat findstr com.lagou.education > E:/apk/logcat.txt
```

筛选指定包      存放路径及日志名称

第二步：



## 2.3.5 adb 其他相关命令

- 1.adb devices , 获取设备列表及设备状态
  - 2.adb connect IP+Port, 无线连接
  - 3.adb get-state , 获取设备的状态    设备的状态有 3 钟
  - 4.adb kill-server , adb start-server 结束 adb 服务, 启动 adb 服务
  - 5.adb logcat查日志
  - 6.adb install , 安装apk, 当前PC端文件所在的路径
  - 7.adb uninstall , 卸载应用, 后面跟的参数是应用的包名
  - 8.adb push , 推送本地文件至 Android 设备 <本地文件目录> <设备文件目录>
  - 9.adb pull , 将 Android 设备上的文件或者文件夹复制到本地 <设备文件目录> <本地目录>
  - 10.adb reboot , 重启 Android 设备
- adb 这个程序自带的一些命令

有三种状态：

- device：设备正常连接
- offline：连接出现异常，设备无响应
- unknown：没有连接设备

## 3 移动端抓包

手机抓包应用场景：

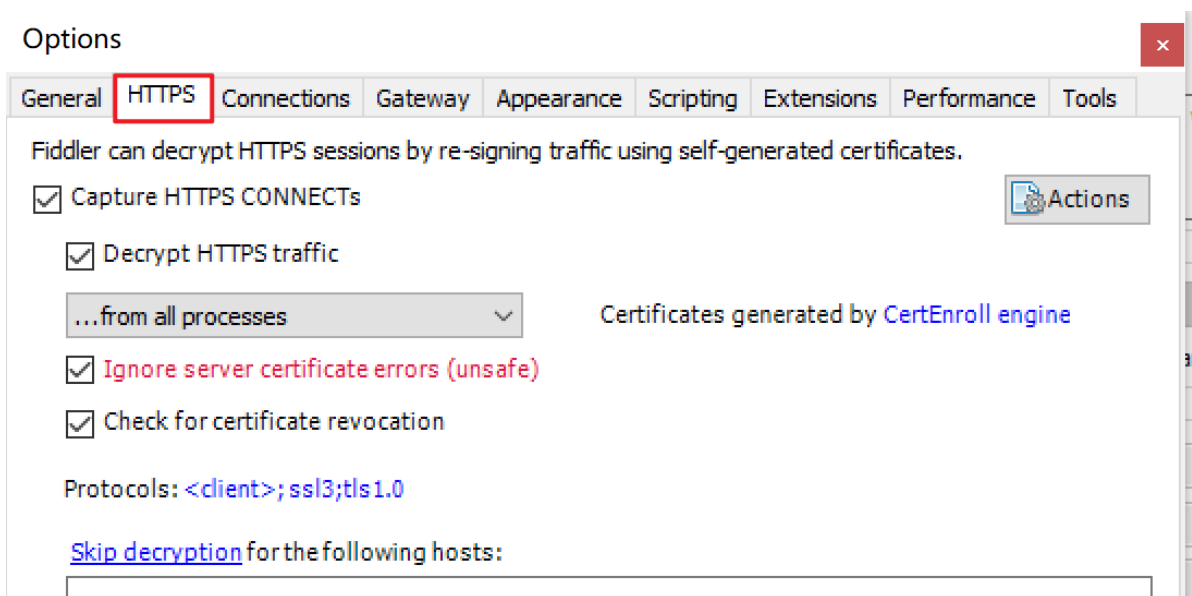
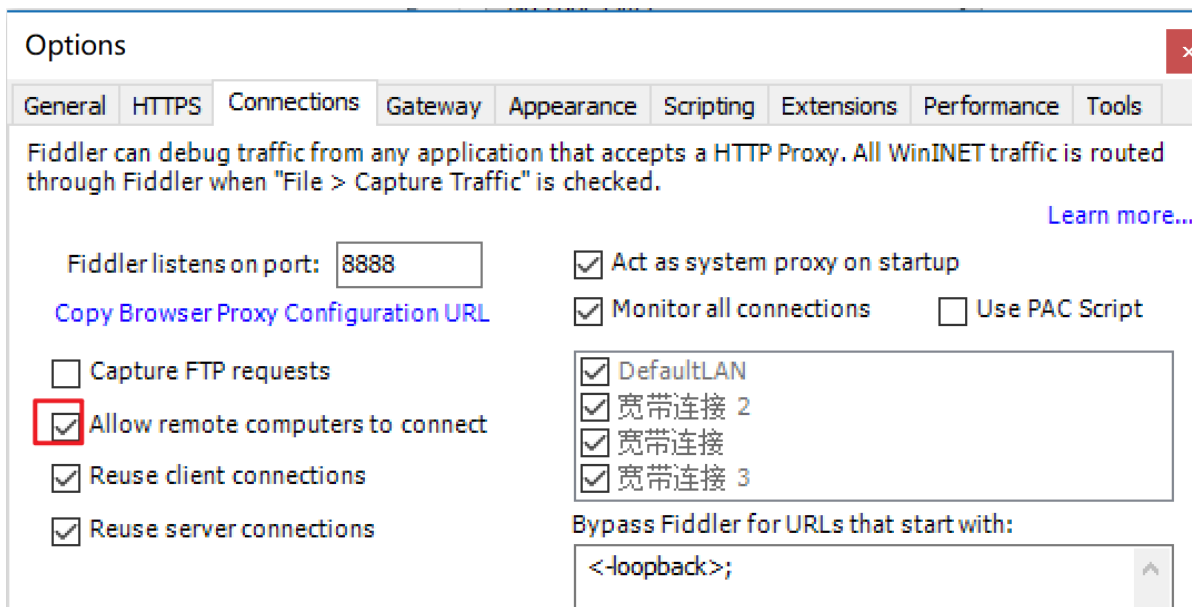
通过手机抓包对数据分析，确定bug产生是前台还是后台

使用fiddler工具对模拟器进行抓包

### 3.1 pc端fiddler配置

配置fiddler允许监听到https（fiddler默认只抓取http格式的）

配置fiddler允许远程连接



## 3.2 对手机端进行参数配置

手机和电脑连接同一个网络

手机系统设置-无线网-点击高级，代理选择手动，主机名就是fiddler的电脑ip地址，端口号，就是fiddler的端口号，与浏览器中ip: 端口号保持一致，点击确定即可

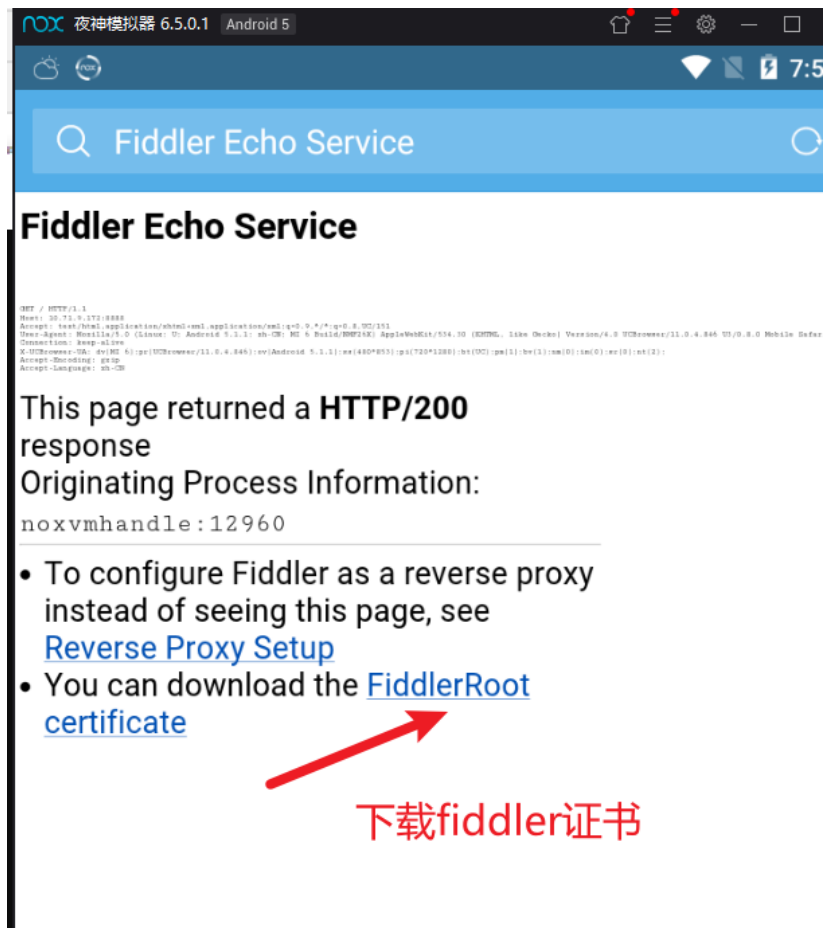
打开手机浏览器 安装fiddler 证书IP:8888

重启后查看

移动端 手动输入IP地址和端口号



浏览器输入IP:端口 举例:192.00.211:8888, 跳转到fiddler下载页面 (如图)



设置 > 更多设置 > 安全 > 凭据下面的 (从SD卡安装)





ios 13 新增步骤

安装fiddler证书 跳转

取消

安装描述文件

安装



## DO\_NOT\_TRUST\_FiddlerRoot

签名者 DO\_NOT\_TRUST\_FiddlerRoot

尚未验证

包含 证书

更多详细信息



打开 设置-通用-关于本机-证书信任设置，将DO\_NOT\_TRUST\_FiddlerRoot 开关打开即可

[← 关于本机](#)

## 证书信任设置

受信任证书存储区版本



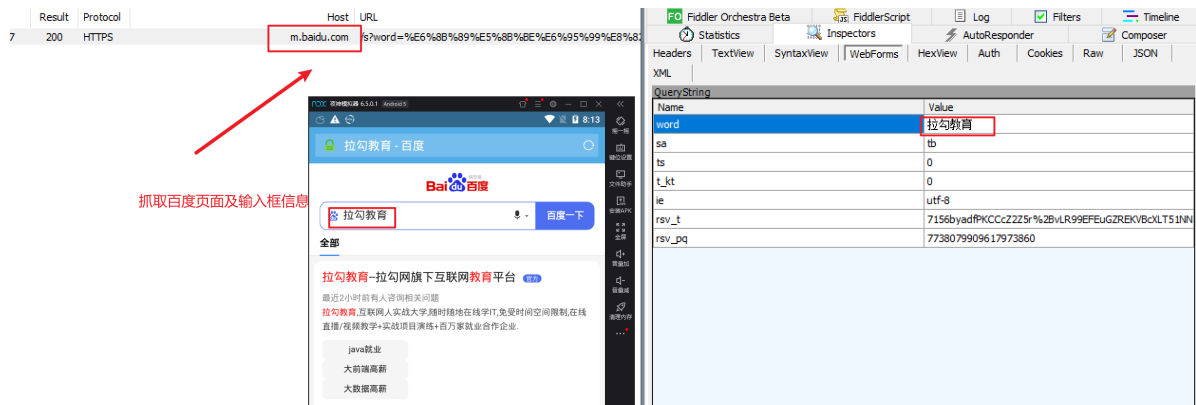
针对根证书启用完全信任

DO\_NOT\_TRUST\_FiddlerRoot



[进一步了解被信任的证书](#)

抓包示例： 百度浏览器 -- 输入框输入‘拉勾教育’



## 4 移动端稳定性工具Monkey

### monkey

monkey 集成在adb工具当中,主要用来做稳定性测试用的,是java语言开发的一款工具,主要用来测试app不会crash(崩溃)的情况

模拟一只猴子来随机操作app,所有的操作都有可能出现

### 4.1 monkey常用的参数

adb shell dumpsys activity | findstr Focused

#### -p 参数

对瑞幸咖啡的app进行随机操作

```
adb shell monkey -p com.flutter_luckin_coffee 1000 (1000次随机事件)
```

#### -v 参数

表示的是记录信息的详细级别

```
level 0: adb shell monkey -p com.flutter_luckin_coffee -v 1000
```

```
level 1: adb shell monkey -p com.flutter_luckin_coffee -v -v 10000
```

最高级别 -v-v-v

```
adb shell monkey -p com.flutter_luckin_coffee -v -v 10000
```

打印出来的信息会更多,会显示出其他程序运行的信息

#### --throttle

用于指定随机事件的间隔时间,单位是毫秒

--throttle ms

```
adb shell monkey -p com.flutter_luckin_coffee -v --throttle 50 100
```

### 忽略崩溃

--ignore-crashes

### 忽略超时

--ignore-timeouts

组合使用：

```
adb shell monkey -p com.flutter_luckin_coffee -v --throttle 50 --ignore-crashes --ignore-timeouts 100 > e:\apk\luckin.txt
```

### 日志分析

如果在日志里面出现了 ANR (application not responding)

如果日志中出现了Exception，可能程序崩溃

### 关于monkey的常见问题

如何重现 ANR 和 crash 过程？

添加参数 -s seed 种子

```
adb shell monkey -p 包名 -s 66 (自定义)
```

-s 是种子数，当执行命令种子数相同事假操作步骤、流程、操作事件视为一致