# 

# Android自动化测试-------Java篇

# 第一章 Android命令介绍

1. ADB-Android debug bridge：电脑和Android设备之间的连接、调试通道
2. Adb常用命令：

注意：在使用adb push和adb pull的时候，要先使用adb remount获取读写权限

1)adb devices：获取连接在当前电脑的所有设备，获取设备id和设备状态

2)adb push/pull：向模拟器中写入文件或者读取文件

3)adb forward：端口转发命令，表示可以将PC端的端口重定向到设备端的端口

4)adb start-server/kill-server：开启adbserver或者杀死server

5)adb install/uninstall：向模拟器中安装或者卸载APK。安装指向的是应用名，卸载指向的是包名

6)adb shell：进入shell命令

7)进入adb shell 后，pm list package:列出当前设备下的全部包名

1. Android命令：可以用于查看当前Android开发环境中相关的系统信息以及操作

1)Android sdk：启动sdk manager管理工具

2)Android avd：打开Android模拟器的管理

3)Android list target：查看当前计算机所有AndroidAPI

4)Android create project：

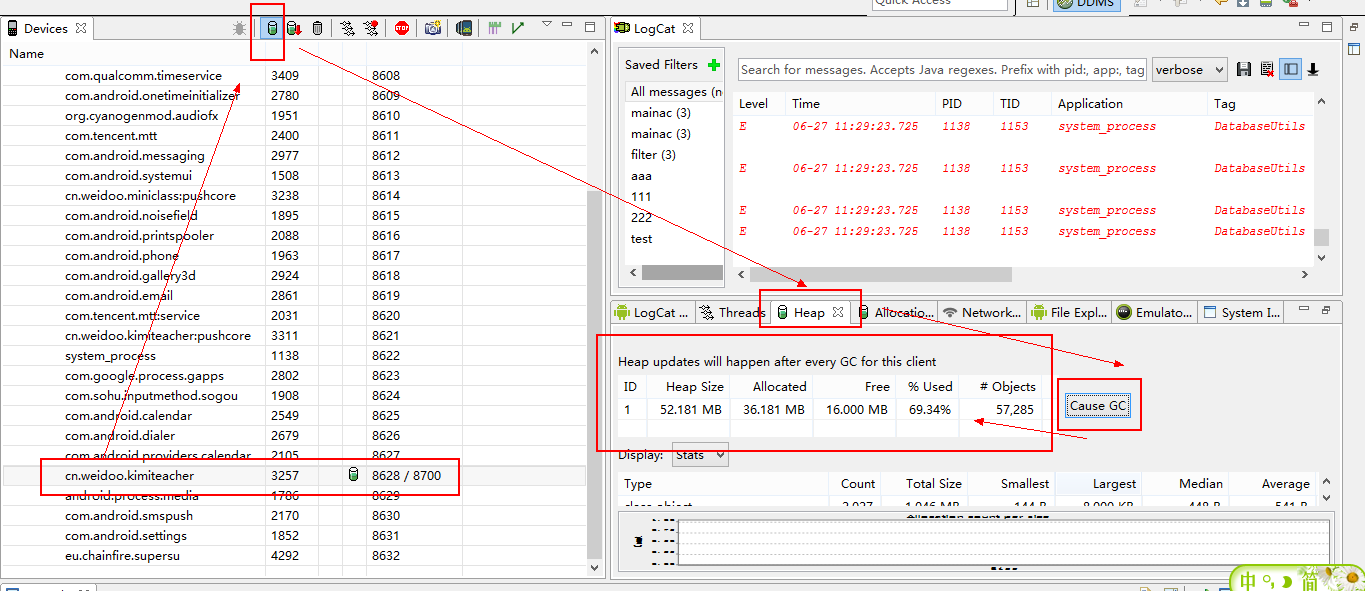
5)android –h:获取帮助

Monitor工具：

1. 定义：集成了在Android开发、自动化测试过程中的对应用调试非常重要的功能。
2. Device
3. Ddms
4. Logcat：

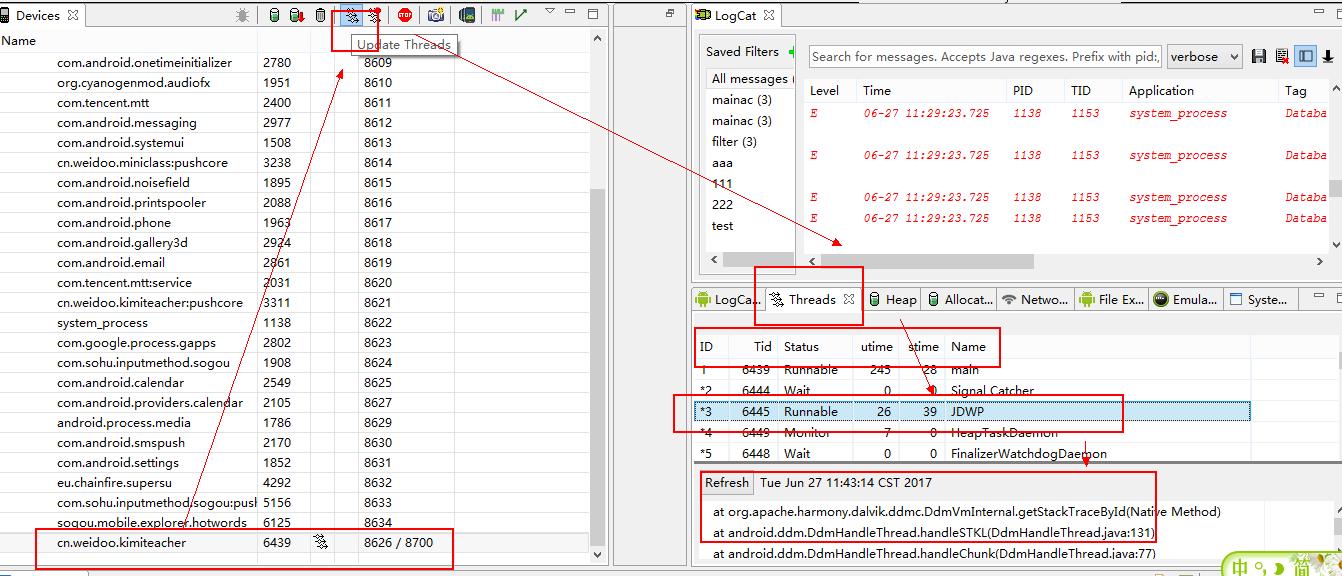
## DDMS：

1. 查看某个进程内存使用状况：



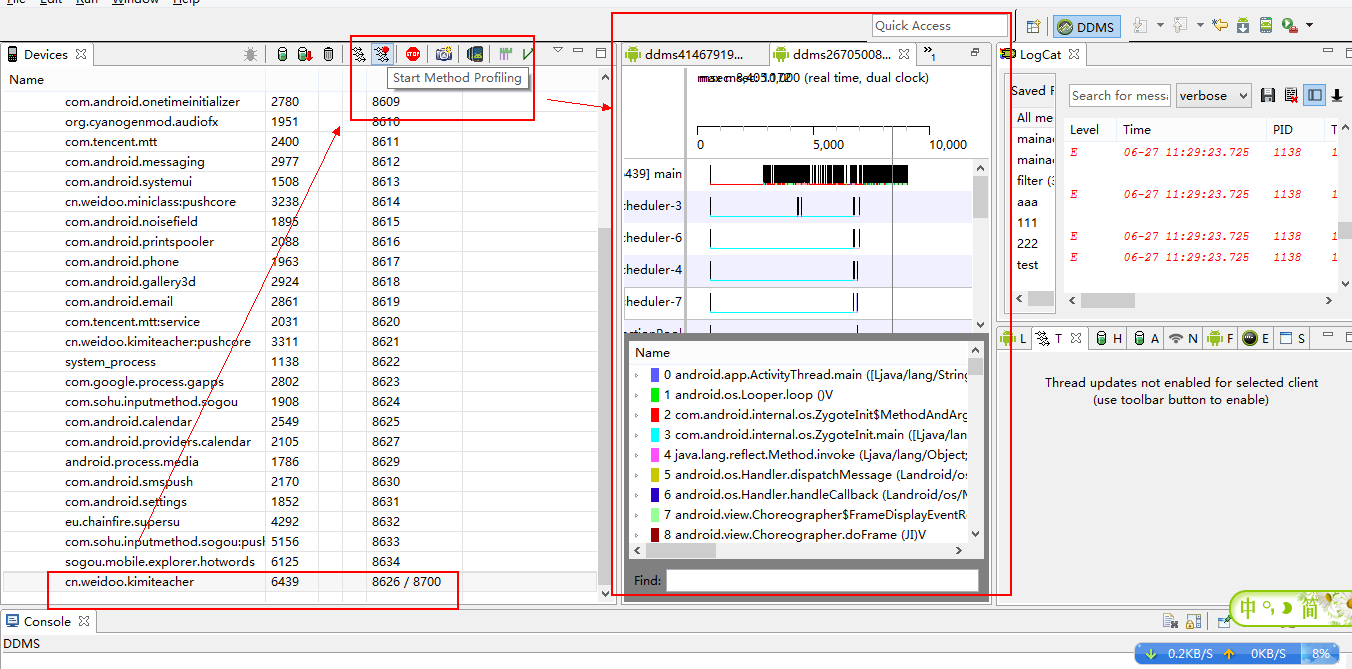
选中进程---选择“update heap” --切换至“heap”---点击“Cause GC”---查看内存使用情况

1. 查看系统各个线程使用的方法：

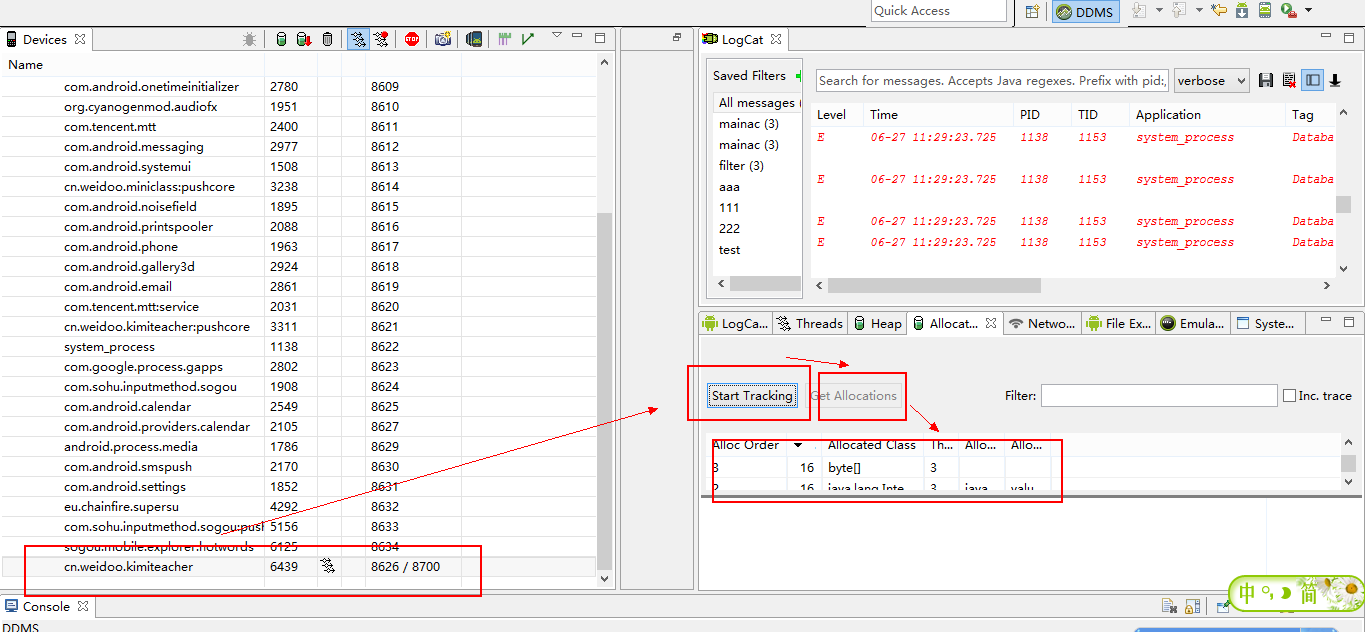


选中进程---“update threads”---切换至“Thread”目录---查看各个进程的使用时间---选择进程--下方可以看到该进程所引用的包

1. 在系统有卡顿的情况下，查看系统的各个方法占用的cpu:

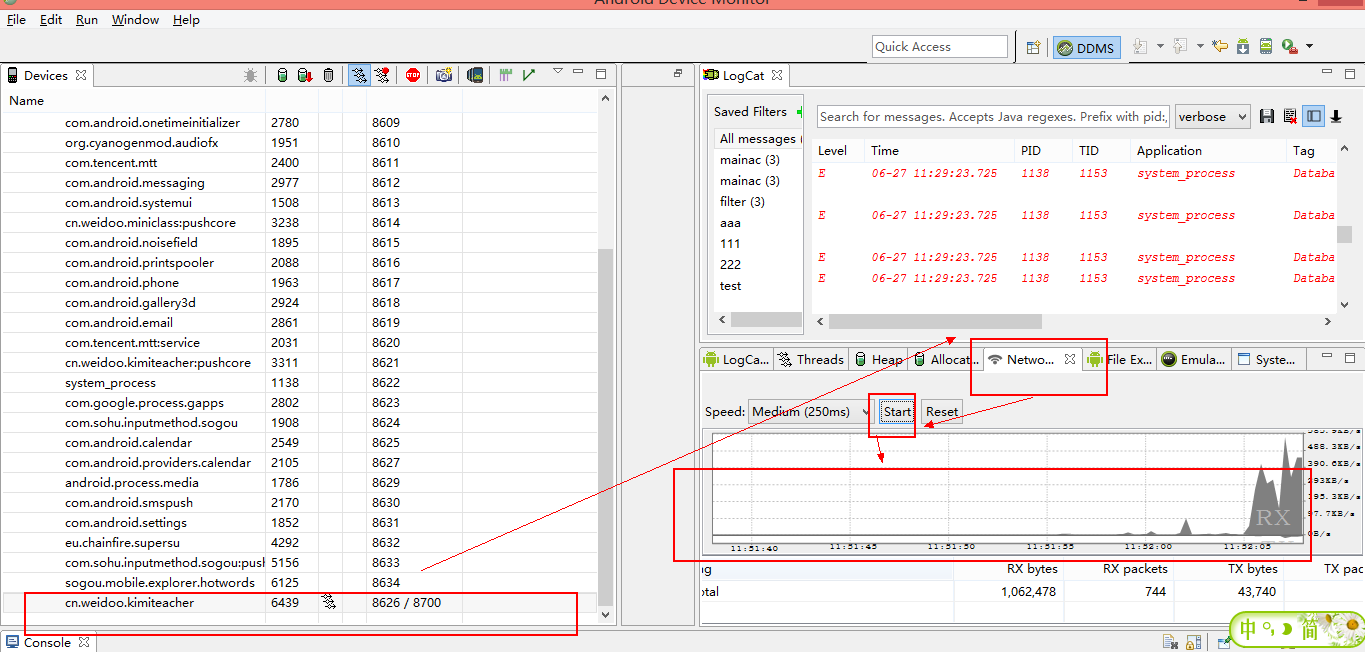


1. 查看系统内存分配使用情况：

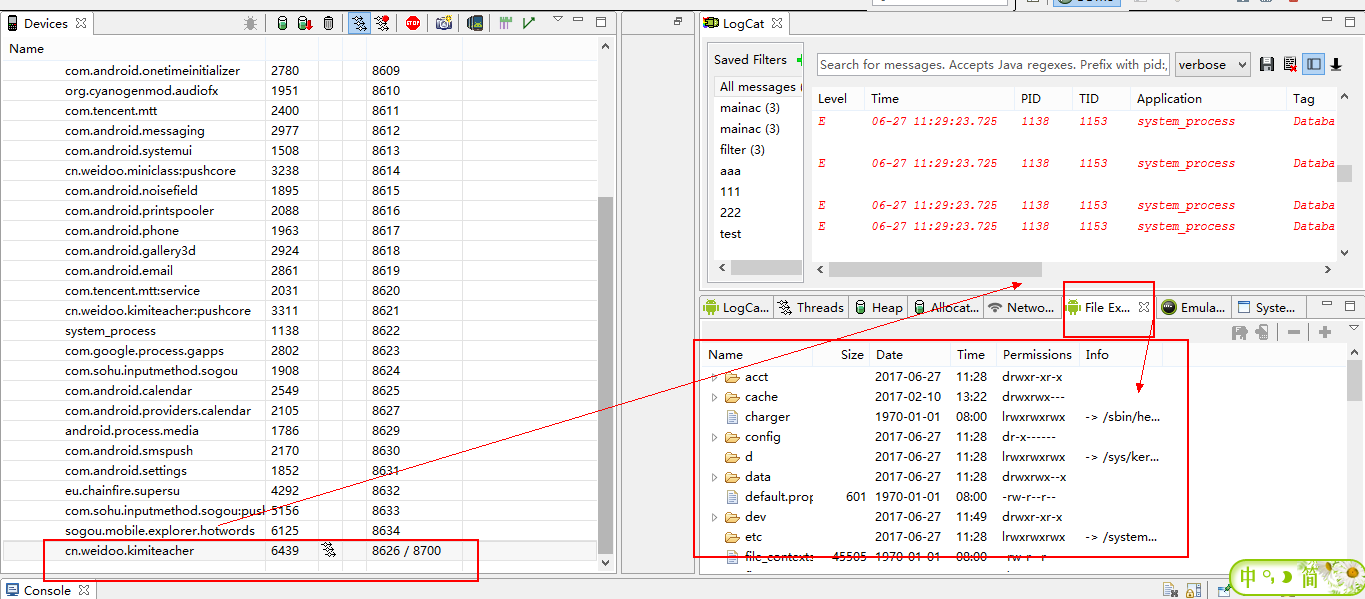


选择进程---“start tracking”--- “get”---下方即可看到进程使用情况

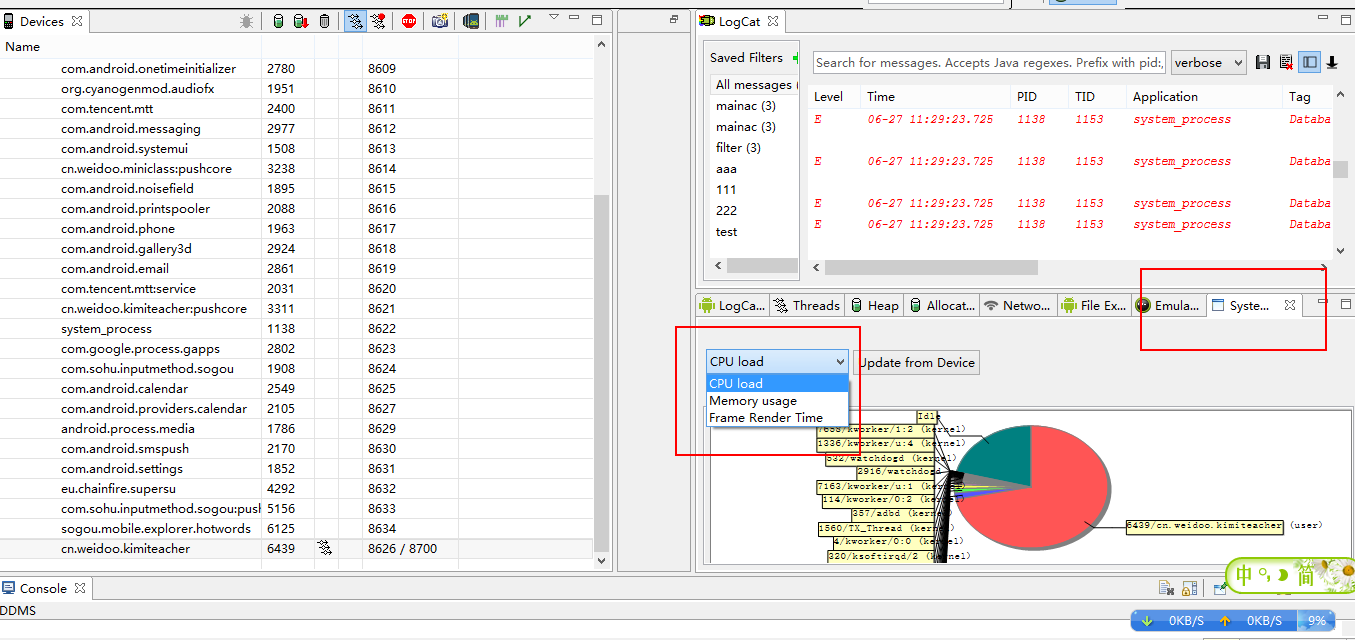
1. 查看进程流量使用情况（RX-接收流量 TX -发送流量）：



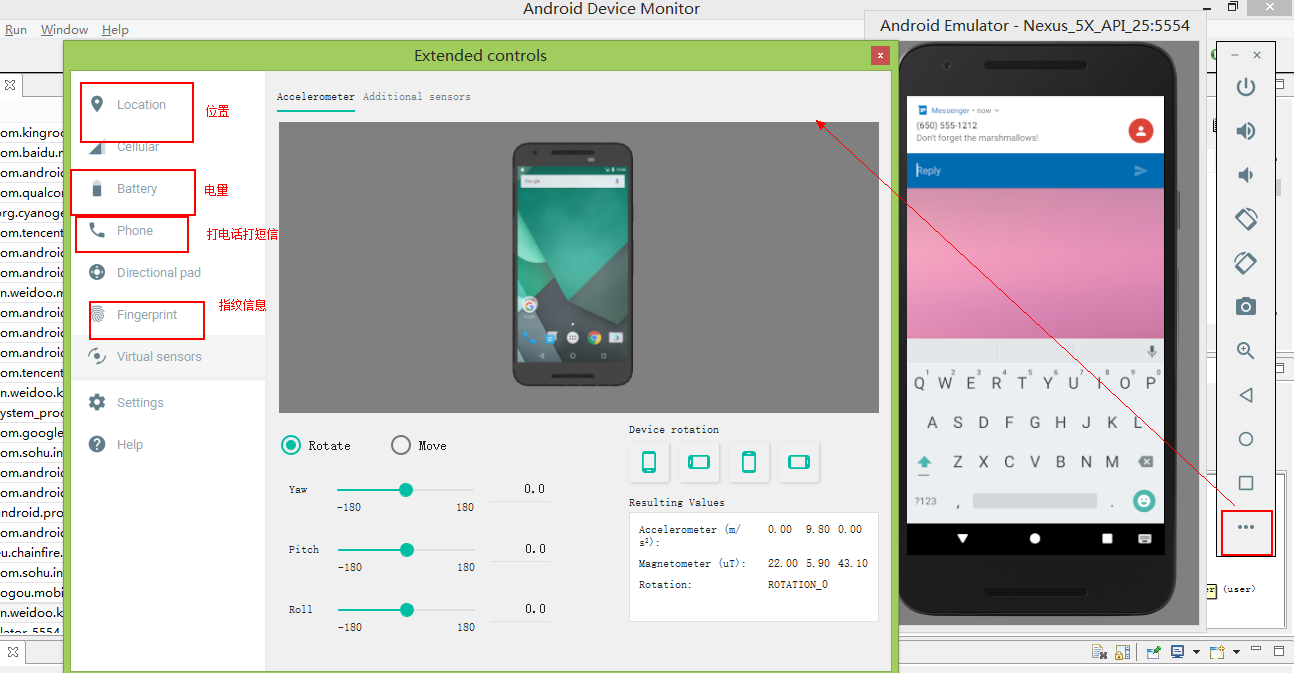
1. 查看当前设备的文件信息



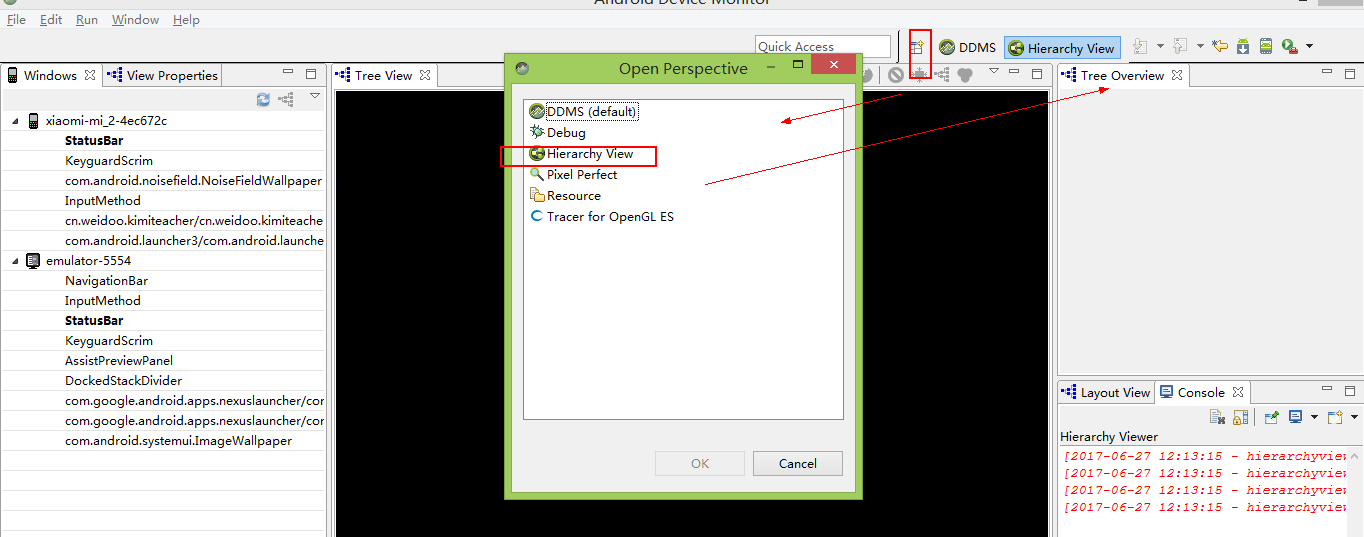
1. 查看系统使用情况：



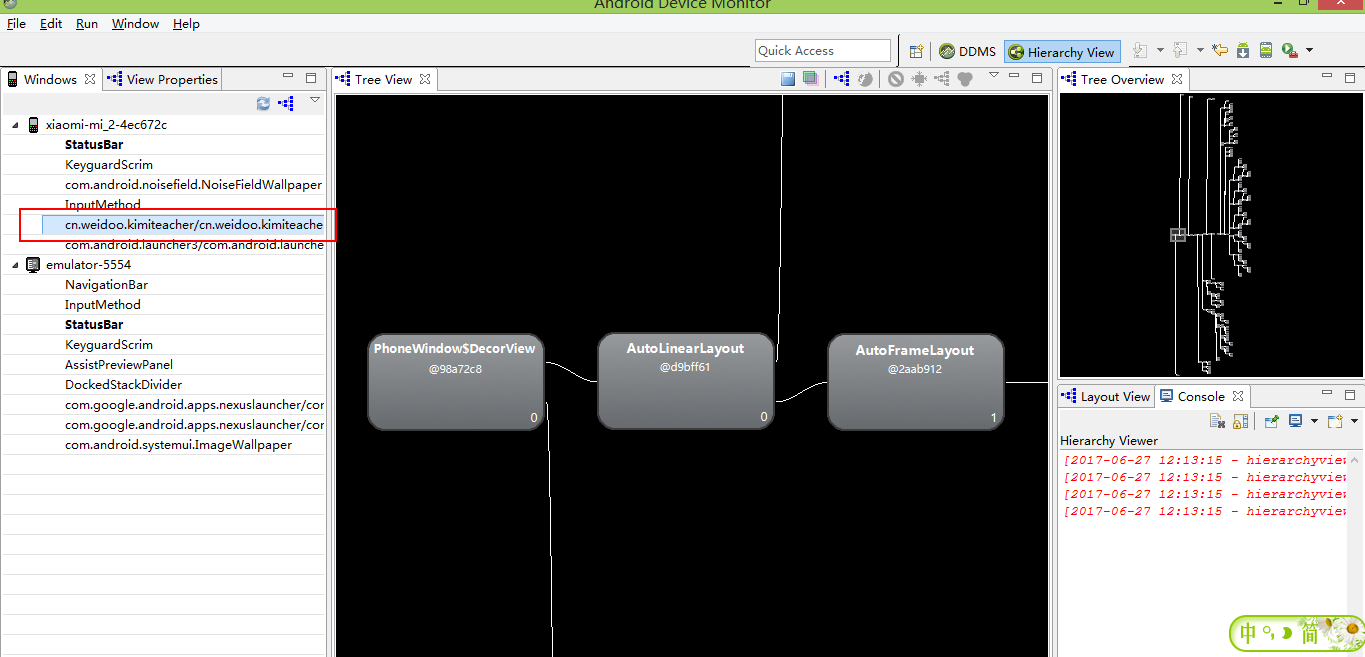
1. 模拟器控制：



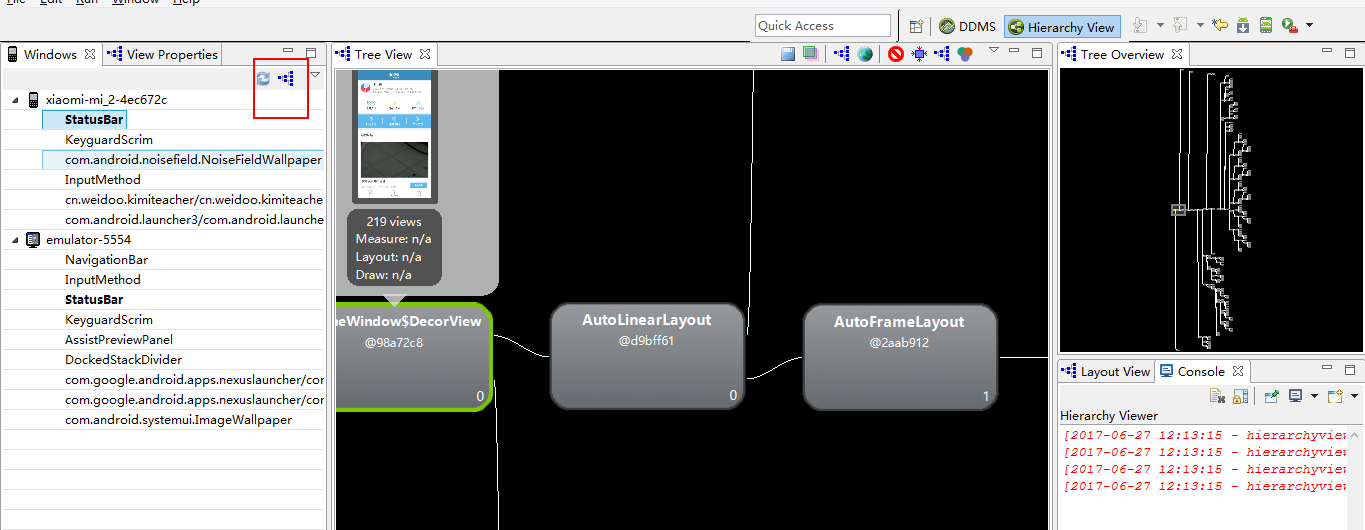
1. 切换视图：



1. 查看进程树状结构：



生成树状结构：



第二章：monkey工具详解

1. monkey：是Android系统中自带的一个黑盒测试工具，一般通过随机出发界面事件，来确定应用是否会发生异常。多用于Android应用的稳定性、压力测试
2. Monkey命令格式：adb shell [options ] <次数>
3. 启动一个应用：LanchActivity(包名，主Activity名)

计算机为例：

Type=raw events

Count=10

Speed=1.0

Start data >>

LanchActivity(com.android.cacultor2.com.android.Cacultor)

#touch 9

DispatchPointer(0,0,,0,500,600,0,0,0,0,0,0,0)

DispatchPointer(0,0,,1,500,600,0,0,0,0,0,0,0)

UserWaiter(1500)

#touch 6

DispatchPress(KEYCODE\_6)

UserWait(1500)

#touch +

DispatchPress(KEYCODE\_PLUS)

UserWait(1500)

#touch 7

Tap(100,800,50)

UserWait(1500)

#touch =

DispatchPress(KEYCODE\_EQUALS)

UserWait(1500)

RatateScreen(2,1)

UserWait(500)

第三章

1. Monkey:是Android系统中自带的一个黑盒测试工具，一般通过随机触发界面事件，来确定应用是否会发生异常，多用于Android应用的稳定性。压力测试
2. Monkey每一个操作都有序列值seed，我们可以通过seed来复现每一次的操作
4. adb shell monkey -v ：指定了monkey输出日志的级别

-s（seed）:指定seed重现测试过程

--throttle 300：延时。如果不指定则不发生延迟

--pct-touch 30 ：触摸时间所占比例

--pct-motion 20 ：移动时间所占比例

--pct-nav 20 ：导航事件

--pct-majornav 15 ：主要导航事件

--pct-appswitch 5 ：覆盖情况

--pct-anyevent 5 ：除了其他时间的比例

--pct-trackball 0 ：轨迹球的占比

--pct-syskeys 0 ：物理按键的时间占比

-p cn.weidoo.kimiparent ：-p指定测试包名

30000

--ignore-crashes

--ignore- timeouts

--ignore-security-exceptions

--ignore-native-crashes

> E:\log\logparent1700.txt

1. monkey命令：

**adb shell monkey -v --throttle 300 --pct-touch 30 --pct-motion 20 --pct-nav 20 --pct-majornav 15 --pct-appswitch 5 --pct-anyevent 5 --pct-trackball 0 --pct-syskeys 0 -p cn.weidoo.kimiparent 3000 --ignore-crashes --ignore- timeouts --ignore-security-exceptions --ignore-native-crashes > E:\log\log1905.txt**

1. 查看对应APP的aapt dump badging e:\weidoo\KimiParent-debug-vdebug20170719(vc3).apk（只在第一行即可查看到包名）

Monkey日志搜索crash、exception、ANI

1. 操作步骤：
2. 1.adb shell查看adb server是否启动(adb server若未启动，大多是因为端口号被占用)
3. 2，输入monkey命令（使用累加次数，五千、一万、五万、十万）

[利用monkey测试android，入门级用户可能遇见的错误及解决办法](http://blog.csdn.net/zm2714/article/details/7977930)

**一、D:\android\android-sdk-windows\tools目录中没有adb.exe**

检查一下你的android-sdk-windows目录下有没有platform-tools目录。并检查访目录下是否有adb.exe

**二、在cmd中执行“adb shell”，则提示“error: device not found”**

没有发现设备。如果用手机设备测试，请用USB数据线与电脑连接，同时并安装好手机驱动；如果你采用模拟器测试，你应该先启动eclipse，然后设置好模拟器，并进入模拟器测试状态，如下图所示

## 三、执行“adb shell”，提示“error: more than one device and emulator”

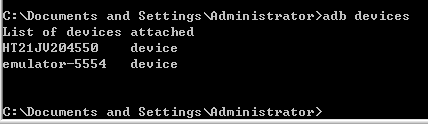
### 第一种情况：确实用多个设备或者模拟器

错误说明是有一个以上的设备和仿真器。这是因为我启用了模拟器，同时我也将我的手机通过USB线连接到了电脑。所以这里存在两个设备。这种情况下，需要指定连接某一个设备或者模拟器。

**1、获取模拟器或设备列表**

命令：**adb devices**

效果如下图所示：



**2、指定device来执行adb shell**

命令：**adb -s devicename shell**

在多个device的时候，执行adb命令一般都需要用参数-s指定device。

例如卸载emulator-5554上的包com.soft.camera，命令如下  
**adb -s emulator-5554 uninstall com.soft.camera**

例如通过monkey测试  
**adb -s emulator-5554 shell monkey -v -p your.www.com 500**

### 第二种情况：确实有一个设备或者模拟器

之所以显示有多个，是因为此设备用过两次。使用命令 adb devices 会看到这个设备有两种状：offline 和 device，有以下两种解决办法：

**1、重启adb.exe服务**

一般办法：在命令窗口输入 **adb start-server** 重新启动adb.exe服务

**2、终极方法，结束adb进程**

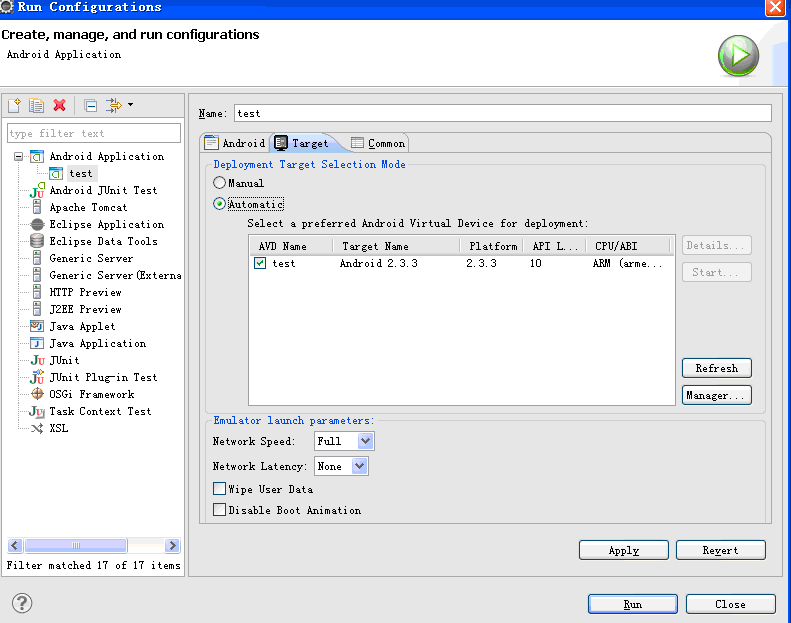
如果上述方法不可以，终极方法：打开任务栏，选择“进程”选项卡，找到adb.exe进程，结束之，重新启动就完事OK啦（或者在命令窗口输入命令 **adb kill-server** ）

## 四、在手机上通过monkey测试

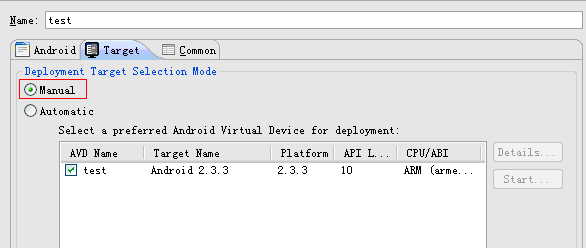
在cmd中输入：**adb -s HT21JV204550 shell monkey -v -p your.www.com 500**

出现：No activities found to run, monkey aborted.

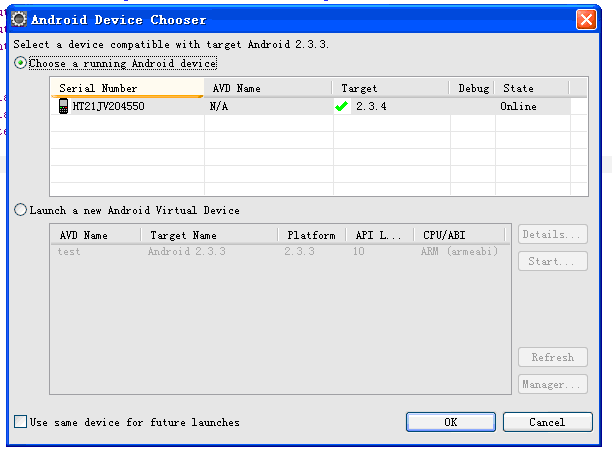
原因：起初我是通过模拟环境测试，Eclipse设置如下：



通过手机测试，应该将“Target”选项卡中由原来的“Automatic”项改为“Manual”，如下图所示：



然后选择“Target”选项卡的前面一个选项卡“Android”，点击下面的“Run”。执行Run后结果如下：



点击OK，程序会将APK写入手机，这时就可以在cmd中输入命令：**adb -s HT21JV204550 shell monkey -v -p your.www.com 500** ，通过在真实手机环境下进行测试了。

## 五、关于monkey命令简要概述

### 一、九个事件及百分比控制

**1、九个事件**

**--pct-touch <percent> 0**  
调整触摸事件的百分比(触摸事件是一个down-up事件，它发生在屏幕上的某单一位置)（——点击事件，涉及down、up）

**--pct-motion <percent> 1**  
调整动作事件的百分比(动作事件由屏幕上某处的一个down事件、一系列的伪随机事件和一个up事件组成) （——注：move事件，涉及down、up、move三个事件）

**--pct-trackball <percent> 2**  
调整轨迹事件的百分比(轨迹事件由一个或几个随机的移动组成，有时还伴随有点击)--（轨迹球）

**--pct-nav <percent> 3**  
调整“基本”导航事件的百分比(导航事件由来自方向输入设备的up/down/left/right组成)

**--pct-majornav <percent> 4**  
调整“主要”导航事件的百分比(这些导航事件通常引发图形界面中的动作，如：5-way键盘的中间按键、回退按键、菜单按键)

**--pct-syskeys <percent> 5**  
调整“系统”按键事件的百分比(这些按键通常被保留，由系统使用，如Home、Back、Start Call、End Call及音量控制键)

**--pct-appswitch <percent> 6**  
调整启动Activity的百分比。在随机间隔里，Monkey将执行一个startActivity()调用，作为最大程度覆盖包中全部Activity的一种方法。（从一个Activity跳转到另一个Activity）

**--pct-flip <percent> 7**  
调整“键盘翻转”事件的百分比。

**--pct-anyevent <percent> 8**  
调整其它类型事件的百分比。它包罗了所有其它类型的事件，如：按键、其它不常用的设备按钮、等等。

红色的数字对应下面百分比对应的数字。比如下图中0：15.0%，表示分配--pct-touch事件15%。测试100次分配15次测试down-up。

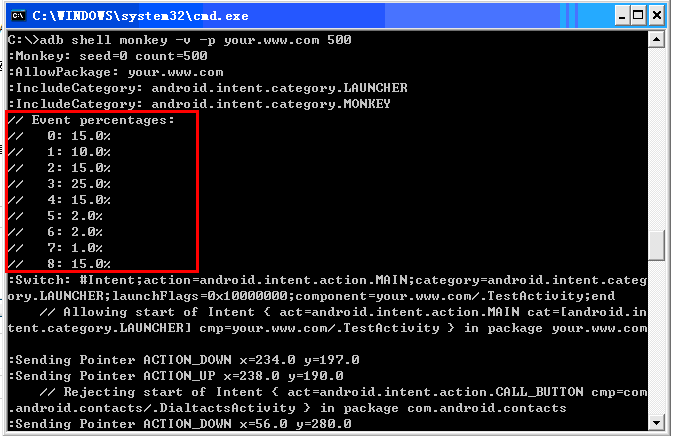
**2、百分比控制**

如果在monkey参数中不指定上述参数，这些动作都是随机分配的，9个动作其每个动作分配的百分比之和为100%，我们可以通过添加命令选项来控制每个事件的百分比，进而可以将操作限制在一定的范围内。

我们先来看一下不加动作百分比控制，系统默认分配事件百分比的情况

命令：adb shell monkey -v -p your.www.com 500

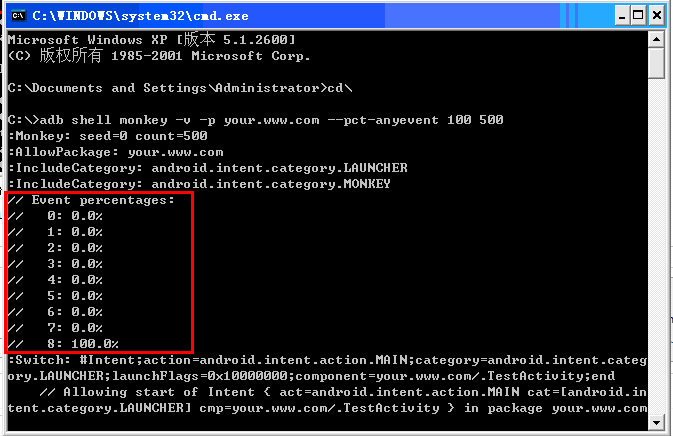
结果：



再看一下指定事件，控制事件百分比之后的情况

命令：adb shell monkey -v -p your.www.com --pct-anyevent 100 500

结果：



说明：--pct-anyevent 100 表明pct-anyevent所代表的事件的百分比为100%。

### 二、约束限制及确定对应包名

**1、约束限制**

-p <allowed-package-name>

1个 -p 选项只能用于一个包。指定多个包，需要使用多个 -p 选项

**2、确定测试包名**

利用monkey测试，通过-p参数确定应用程序在Emulator中所对应的包名。步骤如下：

a、在Eclipse中打开欲测试的项目，并在模拟器中执行。此时，他的APK应该加载到了Emulator（译：仿真器、模拟器）上（

也可以通过命令模式打开模拟器、安装、删除apk

启动模拟器：**emulator -avd test** test为模拟器名称

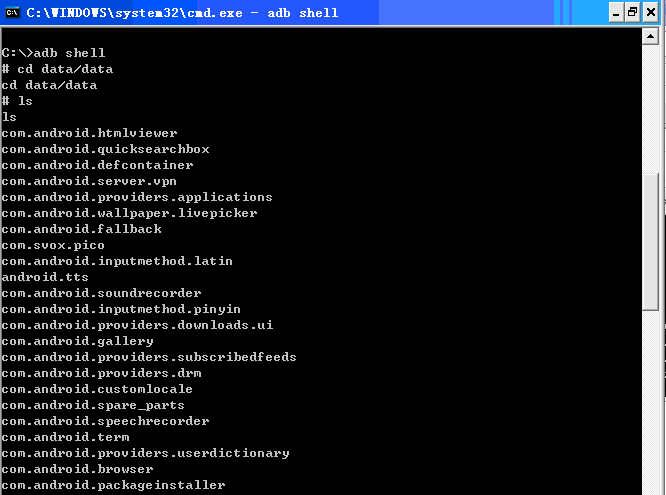
安装：**adb install c:/xxx.apk**

删除：**adb uninstall apk的包名**

）

b、在命令行输入命令：adb shell

c、应用程序包都在data目录中的data目录下，我们输入命令：cd data/data 进入目标data文件夹，在执行命令：ls，可找到刚才加载项目的应用程序包。如下图所示：



找到对应的包名后，就可以通过monkey命令来进行测试了。例如：adb shell monkey -v -p your.www.com 500

随便说一下，最后的500，表示测试触发500个事件（500个事件要按比例分摊到9个事件当中，如何指定不同的事件比例上面已经说过了）。

如果在事件之间插入延时，使用参数：--throttle 5000 这里的5000为毫秒单位。通过这个选项，可以降低monkey的执行速度；如果不指定该选项，Monkey将不会被延迟，事件将尽可能快的被产生。

-s <seed> 伪随机数生成器的seed值。如果用相同的seed值再次运行monkey，它将生成相同的事件序列，对9个事件分配相同的百分比。

-v 命令行的每一个-v将增加反馈信息的级别。

至于其它的一些参数，就百度一下吧，暂时先写到这里。

## 六、在monkey测试中常用的命令组合有

1、monkey -p com.yourpackage -v 500 简单的输出测试的信息。  
2、monkey -p com.yourpackage -v -v -v 500  以深度为三级输出测试信息。  
3、monkey -p com.yourpackage --port 端口号 -v 为测试分配一个专用的端口号，不过这个命令只能输出跳转的信息及有错误时输出信息。  
4、monkey -p com.yourpackage -s 数字 -v 500 为随机数的事件序列定一个值，若出现问题下次可以重复同样的系列进行排错。  
5、monkey -p com.yourpackage -v --throttle 3000 500 为每一次执行一次有效的事件后休眠3000毫秒。

ADB使用常见问题解决方案与ADB的常见使用：

问题：the connection to adb is down,and a server error has occured.you must restart adb and eclipse.

解答：1.在eclipse中选择菜单“help-->check for update”，选择更新ADT.按上面的操作完成后，重新启动Eclipse。

2.在cmd中输入adb的根目录，键入adb kill-server，然后adb---startserver，再重启eclipse

3.在任务管理器中杀死adb.exe进程，然后重启eclipse。如果此时无法杀死adb.exe，则检查是否有豌豆荚在运行，豌豆荚杀死之后重新启动。

4.查看在你安装的SDK路径里面platform-tools文件中是否有adb.exe，如果没有则新建一个文件夹platform-tools，并且把tools中的文件全部复制到platform-tools。

5.安装完手机驱动后，关闭豌豆荚，重启eclipse。豌豆荚等类似的程序与eclipse一般不要同时打开，还可在任务管理器中关掉豌豆荚的进程

6.模拟器有时候很不稳定，有时候要重新全部配置一遍开发环境，或者你删除模拟器重新建一个新的模拟器

7.可能是默认端口被占用，在cmd中键入netstat查看，关掉冲突的端口程序，或者设置新的端口。

8.将platform-tools目录和tools目录加到path环境变量中

9.将eclipse进入DDMS模式，重启ADB和eclipse

10.打开SDK，先下载APIA和Android sdk tools 和 Android 和sdk platform-tools再去配置eclipse

11.开关一下USB调试

12.重启电脑

ADB端口被占用：

Demon not running.starting it now port 5037 \*error:could not install

解答：

1. 在命令行输入：adb devices

Error :could not install \*smartsocket\* listener:cannot bind to 127.0.0.1:5037

1. 查看5037进程号5976：

C:\Users\Administrator>netstat -ano | findstr "5037"

TCP 127.0.0.1:5037 0.0.0.0:0 LISTENING 5976

1. 找到占用端口的应用程序：

C:\user\Administrator>tasklist | findstr “5976”

1. 杀掉进程，若出现下面无法终止的问题，只能去任务管理器找到这个进程手动杀死

C:\user\Administrator>taskkill /f /pid 5976

错误：无法终止PID为5976的进程

原因：拒绝访问

1. 最后再执行adb devices

ADB的使用：

1. 杀死adb之后重启adb：

adb kill -server

Adb devices

1. 安装软件：

Adb devices

Adb -s 设备名称 install 软件路径

1. 进入设备内部：

Adb -s 设备名称 shell

1. 进入手机存储目录：(查看文件 )

#cd /data/data/com.cca.mobilephone/ls -1

二、

1. Monkey脚本

adb shell monkey -f <script> 1

c参考源码：https://android.googlesource.com/platform/development/+/master/cmds/monkey/src/com/android/commands/monkey/MonkeySourceScript.java

1. Monkey脚本命令：

DispatchPointer:按击操作，相当于将手指放置于桌面

DispatchPress[key-code]:发送固定的按键事件

LaunchActivity:启动应用的命令，指定应用的包名

UserWait:让脚本执行暂停等待一段时间

RotateScreen:翻转屏幕

Tap:单击事件

1. monkey测试脚本

count= 1

# 执行次数

speed= 1.0

start data >>

# LaunchActivity的参数为（包名，类名）

LaunchActivity(com.android.calculator2, com.android.calculator2.Calculator)

# 点击3

DispatchPress(KEYCODE\_3)

UserWait(200)

# 点击2

DispatchPress(KEYCODE\_2)

UserWait(200)

# 点击+

DispatchPress(KEYCODE\_PLUS)

UserWait(200)

# 点击9

DispatchPress(KEYCODE\_9)

UserWait(200)

# 点击2

DispatchPress(KEYCODE\_2)

UserWait(200)

# 点击=

DispatchPress(KEYCODE\_EQUALS)

UserWait(200)

1. monkey server工具及优化
2. 启动 monkey server

adb shell monkey --port 1080 &

1. 连接 monkey sever

adb forward tcp:1080 tcp:1080

talent 127.0.0.1 1080 使用talent本机直接启动

1. Monkey Server z主要命令

touch[down|up|move] [x] [y]

Press [keycode]

Sleep

Getvar：获取系统变量

Type：输入字符

# MonkeyRunner

MonkeyRunner 简介：

1、MonkeyRunner：Android SDK中自带的一个黑盒测试工具，在PC端通过 Android PI控制设备的运行

2、支持python脚本，可以实现Monkey无法实现的一些逻辑

3、组成部分：MonkeyRunner MonkeyDevice /MonkeyImage

4、MonkeyRunner的各个函数：（启动：tools:monkeyrunner.bat）

首先引入MonkeyRunner、MonkeyDevice、MonkeyImage

from com.android.monkeyrunner import

1)choice方法：choice(string message,iterable choices,sting title)

方法示例：

MonkeyRunner.choice(‘MonkeyRunner choice’,[‘test1’,’test2’],’imooc’)

2)alert方法：alert(string message,string title,string okTitle);

方法示例：

MonkeyRunner.alert(‘MonkeyRunner alert’,’imooc’,’ok’)

3)help方法：help(string format);

MonkeyRunnner.help(“text”);

示例：将help内容写入到文件中

content=MonkeyRunner.help(“html”);

f=open(“help.html”.”w”)

f.write(content)

F.close()

4)input方法：input(String message,string initialValue,string title,string oKtitile,string canceTitle)

方法示例：

MonkeyRunner.input(‘MonkeyRunner input’,’input str’,’imooc’,’ok’,’cancel’)

5)sleep方法：sleep(float seconds)

MonkeyRunner.sleep(4)。让monkeyrunner睡眠4秒钟

6)waitForConnection(float timeout,String deviceld):超时时间，device的ID。如果不指定的话，则默认连接当前ADB连接的设备

## MonkeyDevice模块功能详解

1.MonkeyDevice：

1. Broadcast Intent(参数)：发送广播
2. drag(tuple strat,tuple end,floatduration,integer steps):拖放操作
3. getProperty(string key):获取系统中的环境变量

Device.getProperty(“display.width”)

Device.getProperty(“diaplay.height”);获取使用设备的分辨率

1. getSystemProperty(string key):adb shell-getprop<key> 获取系统调试变量
2. installPackage(sting path):单元测试
3. instrument(string className,dirctionary args):
4. press(string name,directionary type):发送按键时间

Device.press(“KEYCODE\_7,md.DOWN\_AND\_UP”)

1. reboot(string into):重启设备
2. removePackage(string package):卸载安装包

示例：device.removePackage(“包名”)

1. shell(string cmd):执行shell命令
2. startActivity(string....)：启动命令

示例：启动计算器：

Package=”com.android.calculator2”

Activity=”com.androoid.calculator2.Calculator”

runComponent=package+’/’+activity

device.startActivity(componenet=runComponent)

1. takeSnapshot():获取屏幕截图
2. touch(integer x,integer y,integer type):单击手势操作
3. type(string message):将一组字符串通过press的方法逐个发送到设备上
4. wake():唤醒屏幕
5. 获取安装包的包名：device.shell(“pm list package |grep apk名字（没有.apk）”)

## MonkeyImage模块功能详解

MonkeyImage:(对生成的图像进行处理)

1. 包含：ConvertoBytes(将图像转化为字节码)；getSubImage(获取子图像)；sameAS(对两个MonkeyImage进行比较)；writeToFile(将图像写到文件中，生成一个图片文件)
2. convertToBytes(string format)将当前图像转化为字节码：

示例：

1. getRawPixel(integer x,integer y)获取指定坐标位置的像素值
2. getRawPixelInt(integer x,integer y)将像素数组以整型返回
3. getSubImage(tuple rect)根据给定区域获取子图像
4. sameAS(MonkeyImage other,float percent)比较两个MonkeyImage的相似值
5. writeToFile(string path,string format)将MonkeyImage返回到文件

示例：

from com.android.monkeyrunner import MonkeyRunner,MonkeyDevice,MonkeyImage

device=MonkeyRunner.waitForConnection()

image=device.takeSnapshot()【获取屏幕截图】

image.convertTobytes()

image.getRawPixel(100,100)

image.getRawPixelInt(100,100)

subImage1=image.getSubImage((横坐标起始位置，纵坐标起始位置，宽度，高度))

subImage1.writeToFile(“名字”，”格式”)

subImage2=image.getSubImage((横坐标起始位置，纵坐标起始位置，宽度，高度))

subImage2.writeToFile(“名字”，”格式”)

subImage1.sameAS(submage2,09)

## MonkeyRunner的录制回放工具

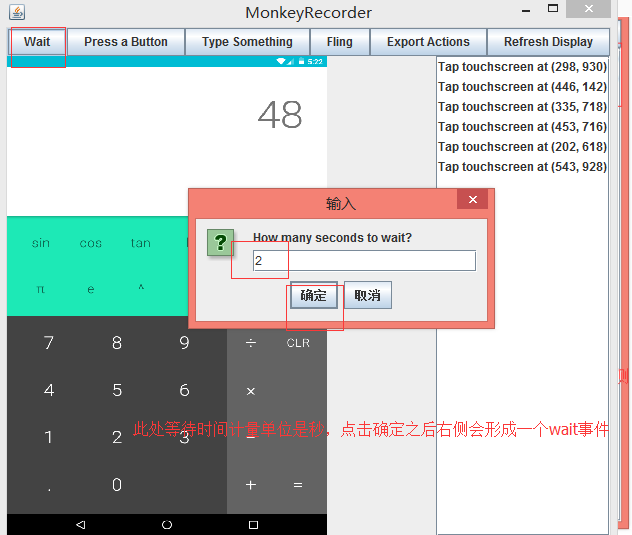
录制；

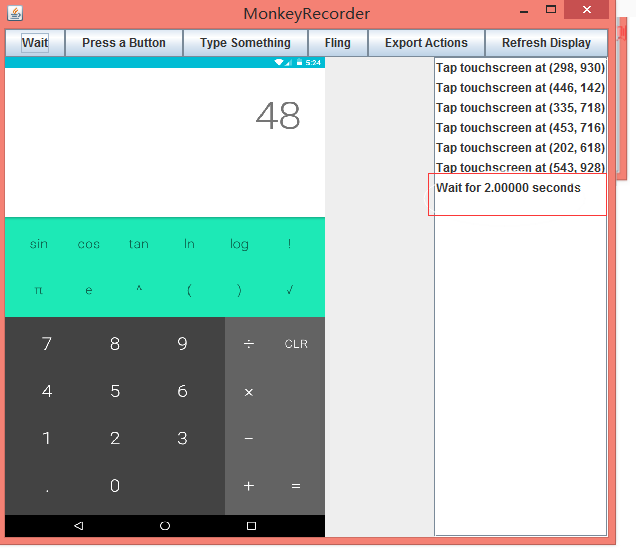
1.在tools 目录下点击monkey\_recorder.py

2.将当前设备的界面同步到工具上，并且在工具上生成相应的操作

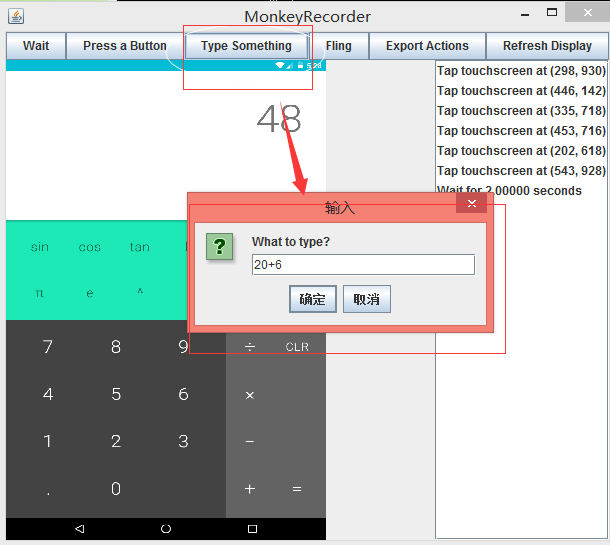
3.

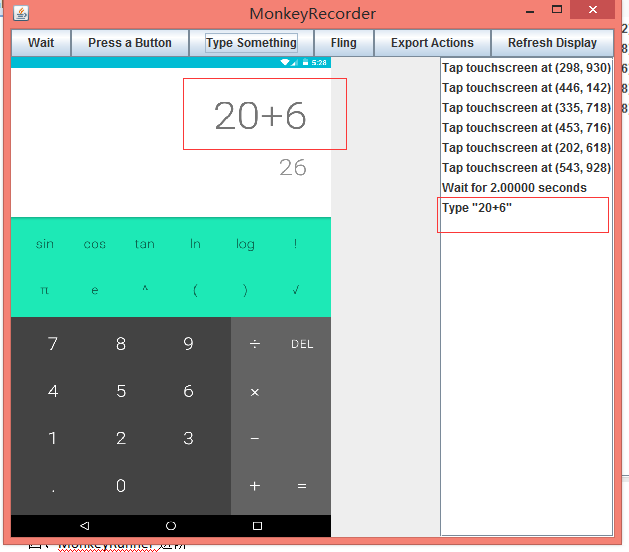
1. 菜单栏的第一个按钮“wait”,点击产生一个对话框，输入想要等待的时间，会在右侧产生一个wait事件。



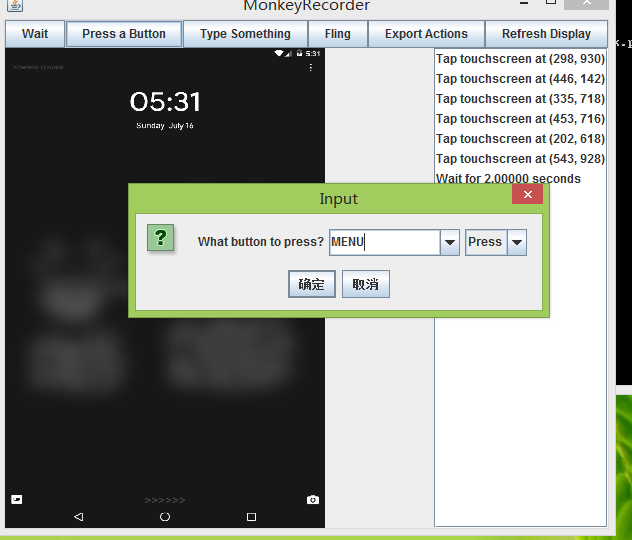


1. Type something 向设备上输入一些字符：



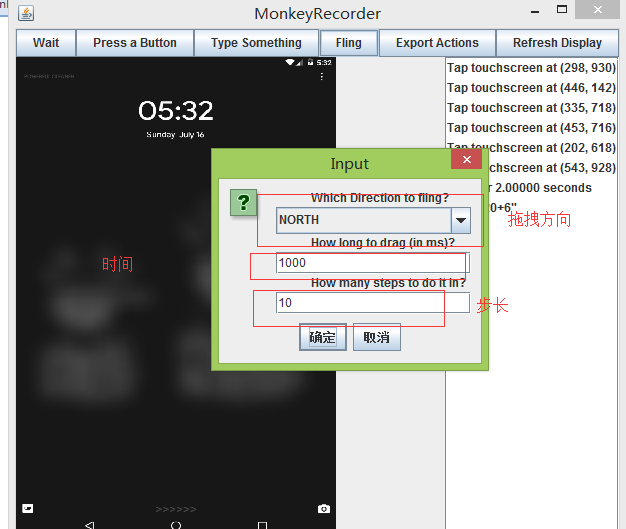


1. Press a button：对页面进行按钮操作：



此按钮为点击APPS按钮

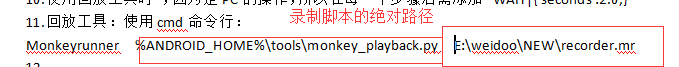
1. Fling：对屏幕做一些滑动、拖拽操作：



向北方向滑动，1s、10步完成

1. Resfresh Display:刷新
2. Export Actions：将所有的操作输出。文件结尾需以“.mr”结尾。
3. 使用回放工具时 ，因为是PC的操作，所以在每一个步骤后需添加“WAIT|{'seconds':2.0,}”
4. 回放工具：使用cmd命令行：

Monkeyrunner %ANDROID\_HOME%\tools\monkey\_playback.py E:\weidoo\NEW\recorder.mr

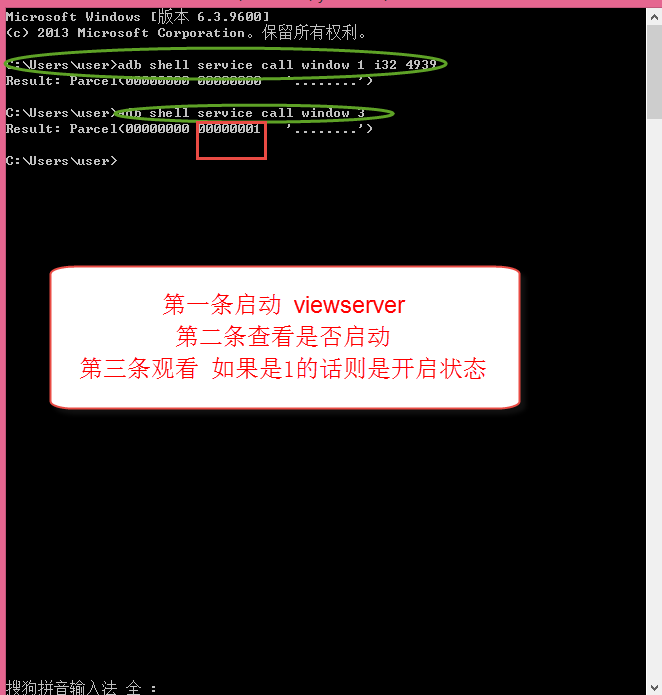
1. 

## MonkeyRunner进阶----EasyMonkeyDevice By:

1. 首先引用EasyMonkeyDevice
2. 打开cmd,输入monkeyrunner
3. 首先：from com.android.monkeyrunner import MonkeyRunner,MonkeyDevice.MonkeyImage
4. 然后：from com.android.monkeyrunner.easy import EasyMonkeyDevice,By
5. device=MonkeyRunner.waitForCinnection() 连接设备
6. device.startActivity(‘com.android.calculator2/.Calculator’) 启动一个Activity 此处是计算器
7. easy=EasyMonkeyDevice(device) 建立一个EasyMonkeyDevice设备



1. 启动viewserver：adb shell service call window 1 i32 4939
2. adb shell service call window 3 查询viewserver的状态

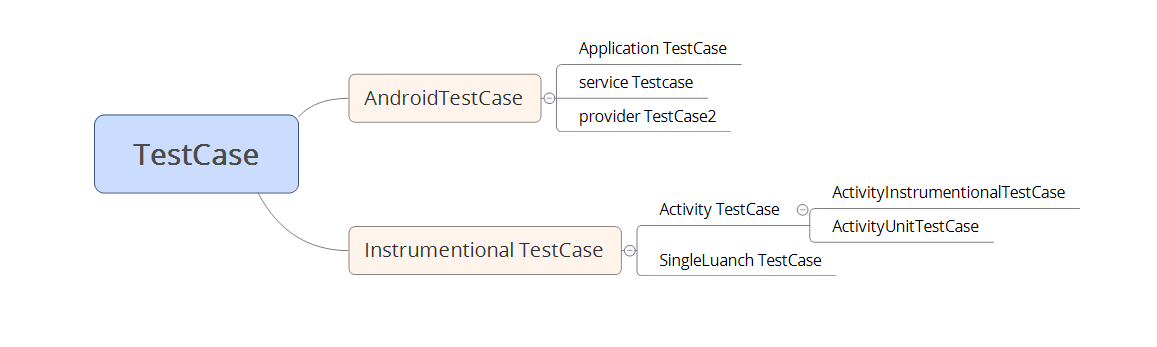


1. 通过easy来进行点击：easy.touch(By.id(‘id/digit\_8’),MonkeyDevice.DOWN\_AND\_UP)
2. 使用tools目录下的hierarchyviewer.bat来进行查看控件的id
3. easy.getFocusedWindowID() 查看当前应用进程的activity名称
4. easy.visible(By.id(‘id/digit7’)) 查看某个按钮是否可见
5. easy.exists(By.id(‘id/digit7’) 判断某个按钮是否存在
6. easy.getText(By.id(‘id/digit7’) 查看某个按钮的text文本域
7. easy.locate(By.id(‘id/digit7’) 查看某个按钮的区域位置
8. hv=device.getHierarchyViewer() 获取HierarchyViewer对象
9. view=hv.findViewById(‘id/digit7’) 获取“7”这个控件的对象信息
10. view.nameProperties.get(‘text:mText’)

第五章 Android单元测试类及其关系

一、

1. Android单元测试类：



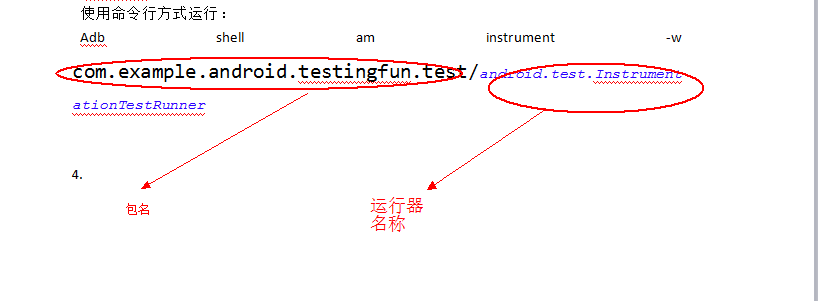
1. Intrumentation 框架及其使用：

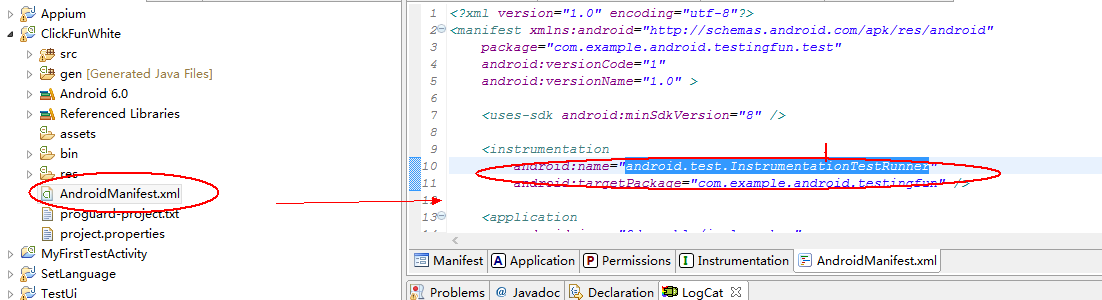
Instrumentation可用来进行黑盒和白盒测试，是谷歌早起推出的UI自动化测试框架，所以向后兼容性最好，可以用来测试安装在低版本的Android操作系统上的APP。但相对来说，此方法只能用来测试单个APP，涉及跨APP交互大的情况此方法不支持。

1. Robotium:

使用命令行方式运行：

Adb shell am instrument -w com.example.android.testingfun.test/*android.test.InstrumentationTestRunner*



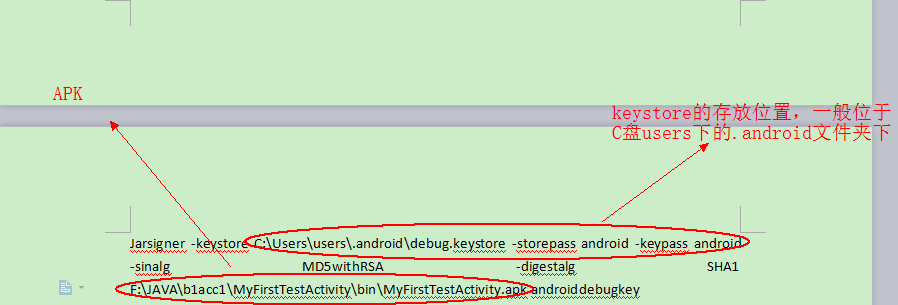
运行器的名称在AndroidManifest.xml中查看：

1. APK重签名：

1.直接使用resign.jar

2.将APK中的META-INF文件夹删掉，使用cmd命令行：

jarsigner -keystore C:\Users\users\.android\debug.keystore -storepass android -keypass android -sigalg MD5withRSA -digestalg SHA1 F:\JAVA\b1acc1\MyFirstTestActivity\bin\MyFirstTestActivity.apk androiddebugkey



1. Robotium主要API

1）点击类

ClickOnText(String text):点击文字控件按钮

clickOnButton(String text)：将按钮上的名称作为参数

clickOnView(View view):使用view对象来点击

clickOnScreen(float x,float y):通过点击屏幕坐标

1. 输入类

enterText(EditText edirText,String text):直接向文本框中输入

ClearText(EditText editText):

1. Get相关操作：

getCurrentActivity()：获取当前的Activity

getText(int index):

getButton(int index):

Getview(String id):

1. search相关操作

SearchaButton:

searchText:

searchEditText:

1. 判断相关操作：

assertCurrentActvity

assertMenmoryNotLow:判断当前内存是否过大

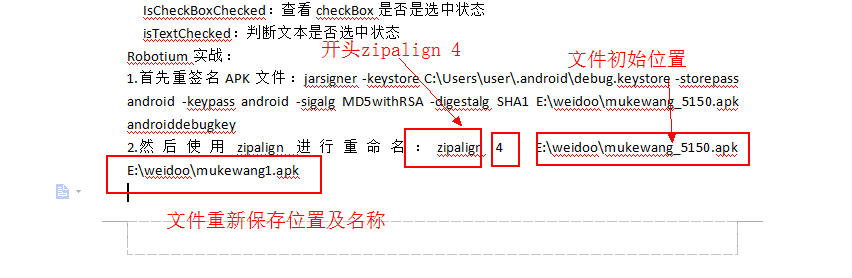
IsCheckBoxChecked：查看checkBox是否是选中状态

isTextChecked：判断文本是否选中状态

# 重签名

Robotium实战：

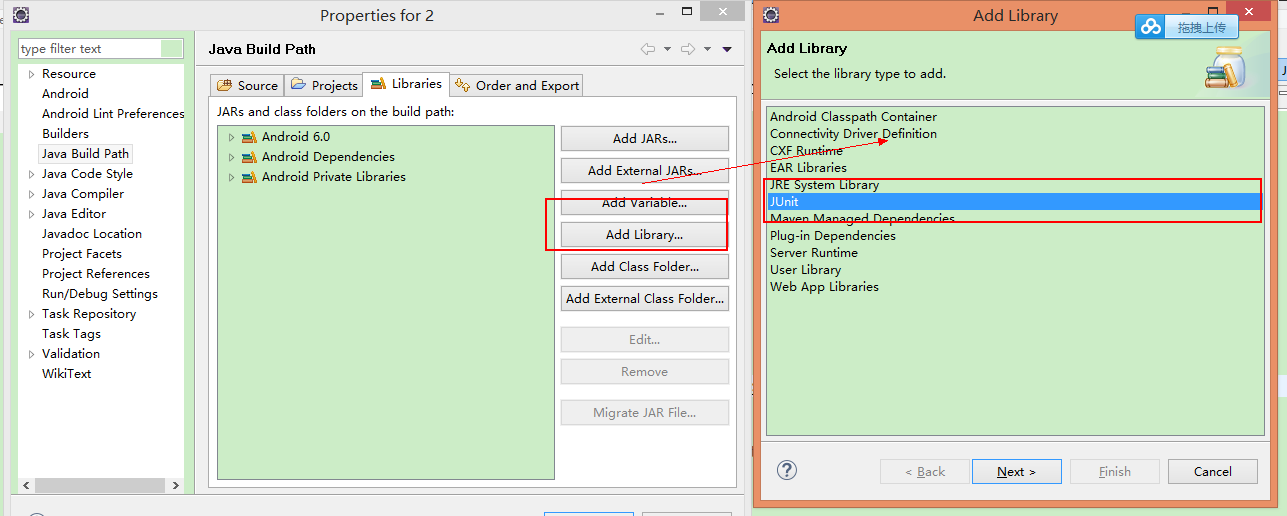
1. 首先重签名APK文件：jarsigner -keystore C:\Users\user\.android\debug.keystore -storepass android -keypass android -sigalg MD5withRSA -digestalg SHA1 E:\weidoo\mukewang\_5150.apk androiddebugkey
2. 然后使用zipalign进行重命名：zipalign 4 E:\weidoo\mukewang\_5150.apk E:\weidoo\mukewang1.apk

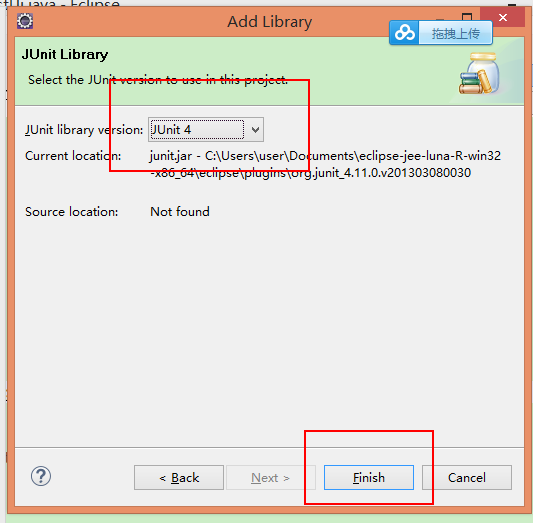


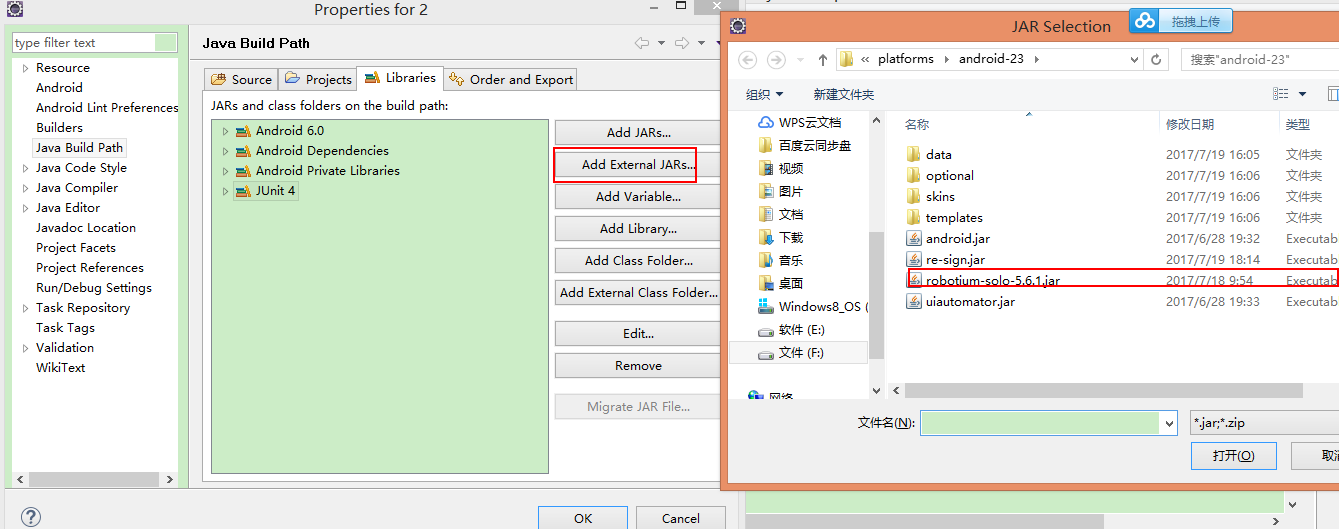
1. adb install E:\weidoo\mukewang1.apk 在模拟器或真机安装
2. 书写测试脚本：

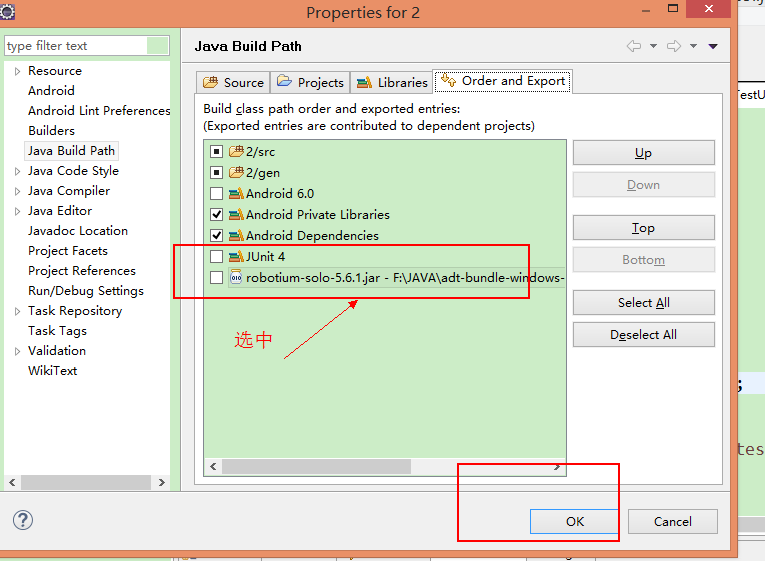
Robotium实战：

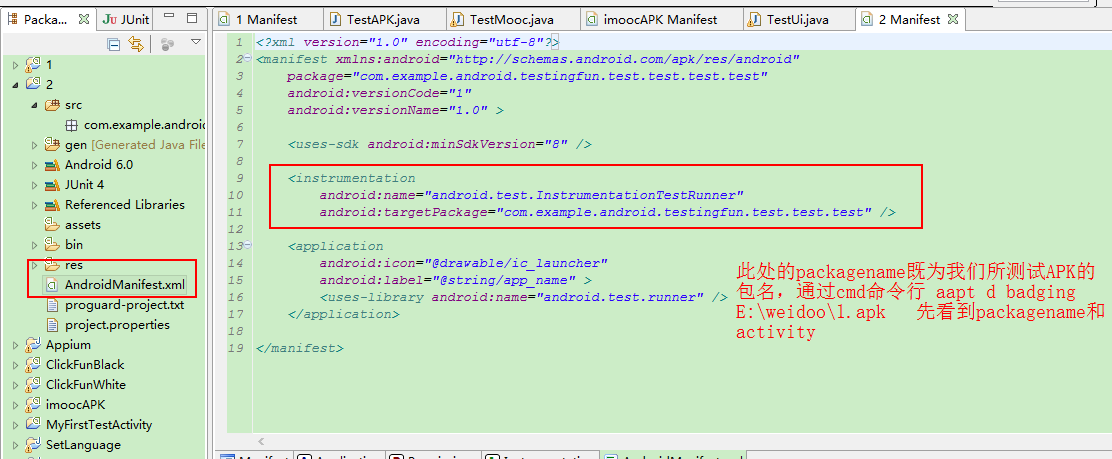
1. 使用cmd命令行重写apk
2. 使用zipalign重写命名
3. 在eclipse中新建一个Android JUnit Text工程：
4. 如果有源码则选择源码所在的project，若没有源码只有apk时则选择“this project”
5. 选择当前模拟器或者真机的API版本
6. 选择build path 先选择add Library，添加JUnit4



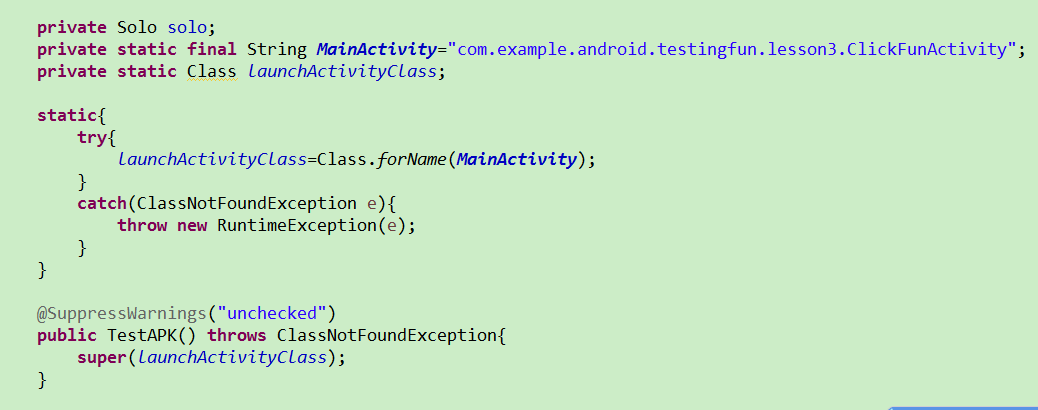


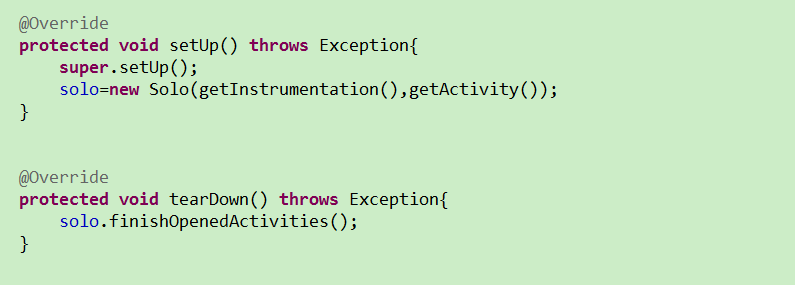




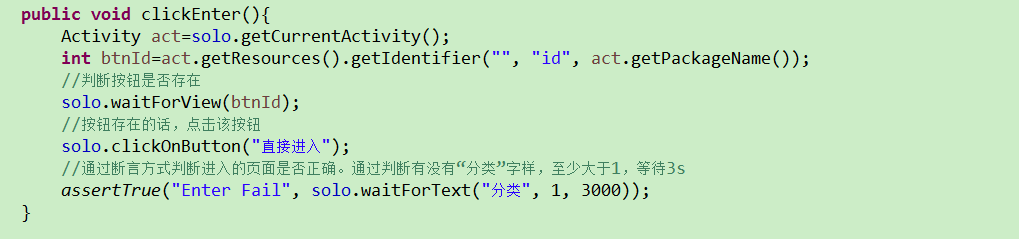


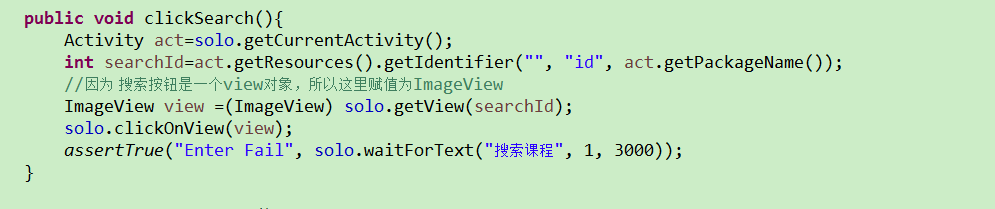
测试类需要继承ActiviyInstrumentationTestCase2<T>。T为待测试的Activity类型，有两个方法值得关注，一个事setUp()方法，这个方法在测试开始的时候会被调用一次。我们可以在这个方法中进行初始化操作。上述代码在这个方法中获取了当前Activity对象的引用。另外一个是tearDown()方法，这个方法在测试结束的时候会被调用一次。如有需要我们可以在这个方法中添加代码。

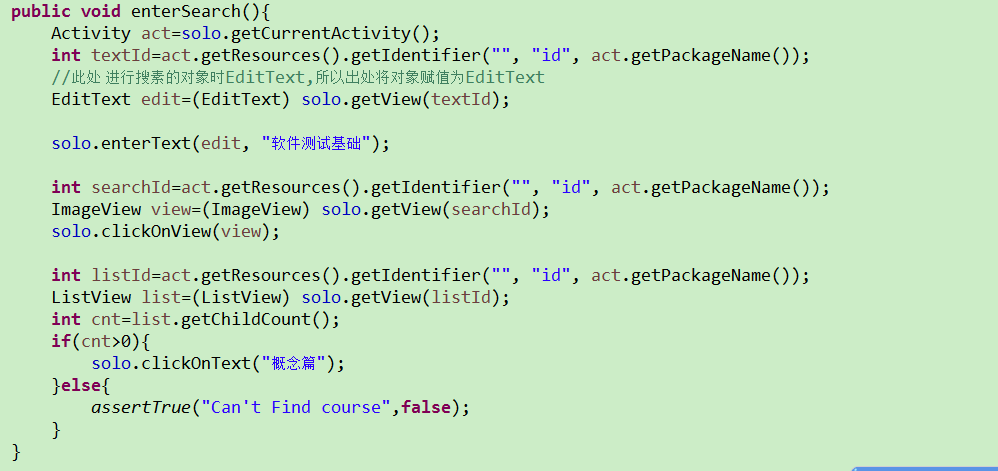


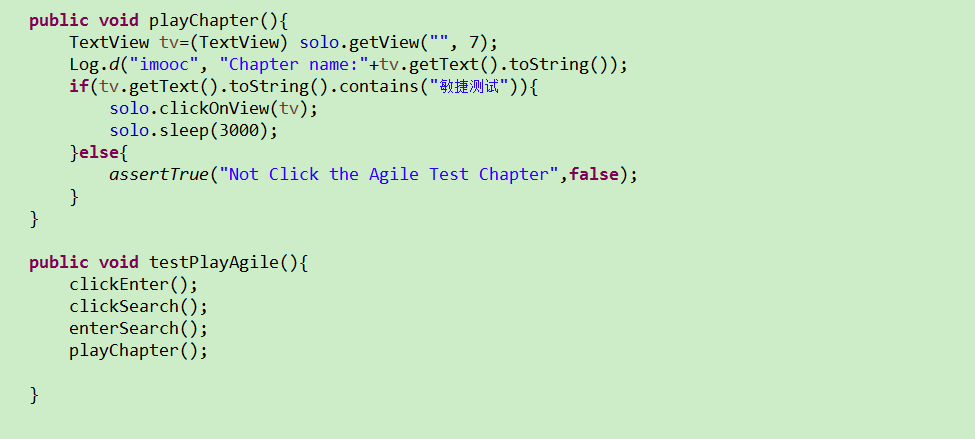


1. 具体实现方法：









白盒测试：



黑盒测试：（参考robotium实战）

Robotium录制和回放工具：

1. 安装

1)点击 Help-install new software

2)输入网址：<http://recorder.robotium.com/updates>

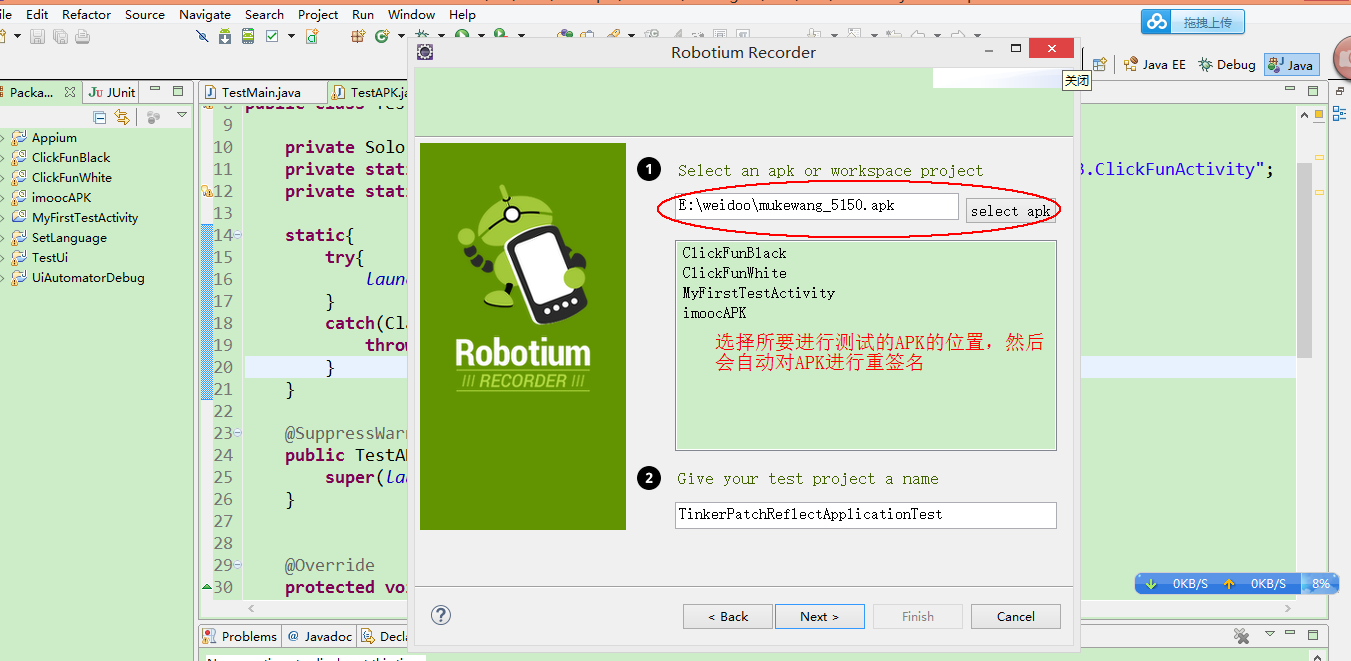
3）点击下载即可

1. 新建一个录制脚本

1）new-other-robotium recoder

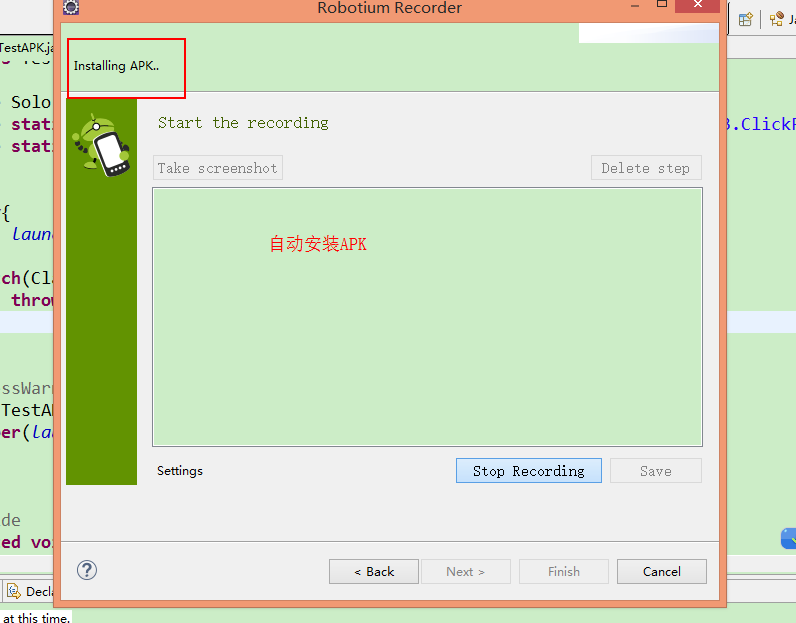
2)选择JDK的位置

3）选择所有进行黑盒测试的APK的位置



此处会自动进行重签名

1. 点击next -自动安装



5）点击stop recoding 则停止录制，点击save则生成新的测试脚本

# 如何安装ADT插件

1. 打开eclipse
2. 点击help---install new software
3. 点击add 名称输入ADT 地址输入：https://dl-ssl.google.com/android/eclipse/
4. 所有的均选择，点击确定，等待

# UiAutomator



## Uiautomator简介：

UIAutomator Viewer 还提供了一个功能，可以查看UIAutomator测试框架可能不支持的控件，只需要在界面上点击“Toggle NAF Nodes”按钮（右上方感叹号标记的按钮）即可查看。如果在使用UIAutomator过程中发现某个控件不能被自动化驱动，可以对照UIAutomator Viewer工具的结果排查问题。

1.优点：

1）API丰富，简单易懂

2）无Acitivity限制，支持跨应用

3）几乎所有的人工操作都能进行模拟

1. 缺点：

1）权限较低，无法像Instrumentation一样获取应用的较高权限

2）有很多的Android API不能使用

3.UiAutomator测试工程的创建和执行：

1）建立一个testUI的工程：

2）继承父类UiAutomatorTestCase

3）写入一个方法testHome() [在UIAutomator测试时，所有方法都必须以test开头]

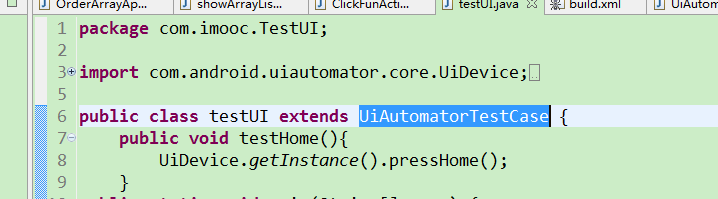
4）打开cmd输入 android create uitest-project -n FirstUi -t 8 -p:f:\java\testui

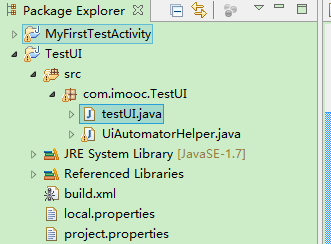
5）通过dir查看是否建立了build.xml文档

6）ant build。形成一个FirstUi.jar。

7）使用adb push将jar包放在待测设备

8）执行命令：adb shell uiautomotor runtest FirstUi.jar -c com.imooc.TestUi.testUI#testHome





1. UIAutomator的快速调试：

UiAutomatorHelper开源调试工具：https:\\github.com\fan2597\UiAutomatorHelper

1）在eclipse中引入下载的源码

2）将UiAutomatorHelper.java 拷贝到Testui下。

3）在testui下创建main函数：

4）

1. 执行

## UiAutomator API

1)UiDevice：提供和设备相关的控制功能，物理按键、坐标点击

2)UiSelect：针对控件元素进行定位。可以属性叠加

3)UiObject：具体的对象，点击，拖拽、获取属性

4)UiCollection： 一组控件的集合，枚举。是对象的集合，可以从集合中进一步查询需要的对象。

5)UiScrollable：针对滚动控件可处理可滚动控件的相关操作

1. UiDevice对象功能详解：

代表设备状态。在测试中，可以通过UIDevice实例来检测设备的各种属性，例如当前的屏幕方向及屏幕尺寸。同时还可以通过UIDevice实例来执行设备级别的操作，例如把设备设置为横屏或者竖屏或者按下home键

**public** **void** testDemo() **throws** RemoteException{

UiDevice device=UiDevice.*getInstance*();

device.pressHome();//点击主页

device.waitForIdle(2000);

device.pressMenu();//点击菜单

device.waitForIdle(2000);

device.pressRecentApps();//列出最近使用APP

//在文本框中输入字符（首页的搜索框）

/\*device.pressKeyCode(KeyEvent.KEYCODE\_I);

device.pressKeyCode(KeyEvent.KEYCODE\_M);

device.pressKeyCode(KeyEvent.KEYCODE\_O);

device.pressKeyCode(KeyEvent.KEYCODE\_O);

device.pressKeyCode(KeyEvent.KEYCODE\_C);\*/

//截图

//device.takeScreenshot(new File("/sdcard/a.png"));

/\*

\* 查看图片：

\* 1.adb shell

\* 2.cd sdcard

\* 3.ls（可查看到a.png）

\* 4.exit

\* 5.adb pull /sdcard/a.png a.png

\*/

//坐标的点击

//device.click(300, 900);

//电源键的灭屏或唤醒按钮

/\*if(device.isScreenOn()){//判断屏幕是否亮屏

device.sleep();

}

device.waitForIdle(2000);

if(!device.isScreenOn()){

device.wakeUp();

}\*/

//按键的拖拽功能

/\*device.drag(400, 900, 250, 500, 50);

device.waitForIdle(2000);

device.drag(250, 500, 400, 900, 50);\*/

//device的滑动操作

/\*device.swipe(1200, 500, 200,1500, 30);

device.waitForIdle(2000);

device.swipe(200, 500, 1200,1500, 30);\*/

//完成通知栏的下拉

//device.openNotification();

//打开快速设置页面

//device.openQuickSettings();

1. }UiSelector详解：

代表一个搜索UI控件的条件，如果发现多个满足条件的控件则会返回第一个控件，返回的结果为UiObject对象。在构造Uiselector的时候可以组合使用多个属性来定位具体的控件。如果没有找到控件会抛出异常

**public** **void** testSelector() **throws** UiObjectNotFoundException{

//text 对应UiAutomator中的text

/\*UiSelector s=new UiSelector().text("Browser");//完全匹配

UiSelector s=new UiSelector().textContains("ser");//包含某种字符串

UiSelector s=new UiSelector().textMatches(".\*er.\*");//匹配正则表达式

UiSelector s=new UiSelector().textStartsWith("B");//以某个字符串开头

\*/

//description 对应uiautomator 中的desc

/\*UiSelector s=new UiSelector().description("Apps");//完全匹配

UiSelector s=new UiSelector().descriptionContains("ser");//包含某种字符串

UiSelector s=new UiSelector().descriptionMatches(".\*er.\*");//匹配正则表达式

UiSelector s=new UiSelector().descriptionStartsWith("B");//以某个字符串开头

\*/

//packageName

//UiSelector s=new UiSelector().packageName("com.android.deskclock");

//UiSelector s=new UiSelector().className("android.widget.TextView").index(3);

/\*若页面中的控件的className多个APP相同时 ，使用叠加属性，查看TextView下的index属性

\* index对应的是同一级空间数下的下标

\* instance对应的是该页面从顶开始的下标

\* UiSelector s=new UiSelector().className("android.widget.TextView").instance(3);

\*/

//UiSelector s=new UiSelector().resourceId("com.android.calculator2:id/digit\_9");

/\*

\* UiSelector s=new UiSelector().focused(true);

\* UiObject browserbtn=new UiObject(s);

\* browserbtn.setText("imooc");

\* 判断当前按钮是否可选中,比如首页的搜索框，输入文本文字“imooc”

\*\*/

/\*

\* 此方法使用ViewGroup来查找，然后使用childSelector进行定位 点击

UiSelector s=new UiSelector().className("android.view.ViewGroup").index(1).childSelector(new UiSelector().index(3));

UiObject browserbtn=new UiObject(s);

browserbtn.click();

\*/

}

UiCollection: 代表控件的集合。获取UICollection的方式和UIObject一样，通过UiSelector查找。UICollection对应Android系统中的ViewGroup以及子控件

**public** **void** testCollection() **throws** UiObjectNotFoundException{

//此viewGroup为当前所选中空间的整体集合（数字）

UiCollection viewGroup = **new** UiCollection(**new** UiSelector().resourceId("com.android.calculator2:id/pad\_numeric"));

UiObject btn9=viewGroup.getChildByText(**new** UiSelector().className("android.widget.Button"), "9");

UiObject btn8=viewGroup.getChildByText(**new** UiSelector().className("android.widget.Button"), "8");

UiObject btn7=viewGroup.getChildByText(**new** UiSelector().className("android.widget.Button"), "7");

UiObject btnEqu=viewGroup.getChildByText(**new** UiSelector().className("android.widget.Button"), "=");

//此viewGroup为当前所选中空间的整体集合（乘除法）

UiCollection viewGroup2 = **new** UiCollection(**new** UiSelector().resourceId("com.android.calculator2:id/pad\_operator"));

UiObject btnTime=viewGroup2.getChildByDescription(**new** UiSelector().className("android.widget.Button"), "times");

btn7.click();

btnTime.click();

btn8.click();

btn9.click();

btnEqu.click();

**int** cnt1= viewGroup.getChildCount();//查看对象个数

System.***out***.println("numpad Count:------"+cnt1);

**int** cnt2 = viewGroup2.getChildCount(**new** UiSelector().className("android.widget.Button"));

System.***out***.println("operator Count:-------"+cnt2);

}

1. 测试实例：

**public** **class** SetLanguage **extends** ~~UiAutomatorTestCase~~ {

**public** **void** testsetChinese() **throws** ~~UiObjectNotFoundException~~ {

// **TODO** Auto-generated constructor stub

~~UiObject~~ app=**new** UiObject(**new** UiSelector().description("Apps"));

app.clickAndWaitForNewWindow();

~~UiObject~~ setting=**new** UiObject(**new** UiSelector().text("Settings"));

setting.clickAndWaitForNewWindow();

~~UiScrollable~~ scroll= **new** UiScrollable(**new** UiSelector().resourceId("com.android.settings:id/dashboard"));

~~UiObject~~ languageSet=scroll.getChildByText(**new** UiSelector().className("android.widget.FrameLayout"), "Language & input",**true**);

languageSet.clickAndWaitForNewWindow();

~~UiObject~~ lang=**new** UiObject(**new** UiSelector().text("Language"));

lang.clickAndWaitForNewWindow();

~~UiScrollable~~ langList=**new** UiScrollable(**new** UiSelector().className("android.widget.ListView"));

~~UiObject~~ chinese=langList.getChildByText(**new** UiSelector().className("android.widget.TextView"), "中文 (简体)", **true**);

chinese.clickAndWaitForNewWindow();

//UiCollection acr\_bar=new UiCollection(new UiSelector().resourceId("android.widget.TextView"));

~~UiObject~~ title=**new** UiObject(**new** UiSelector().text("android.widget.TextView"));

**if**(title.getText().matches(".\*输入法.\*")){

System.***out***.println("ok");

}

}

**public** **static** **void** main(String[] args) {

String jarName="SetLanguage";

String testClass="com.imooc.lang.SetLanguage";

String testName="testsetChinese";

String androidId="6";

**new** UiAutomatorHelper(jarName, testClass, testName, androidId);

}

}

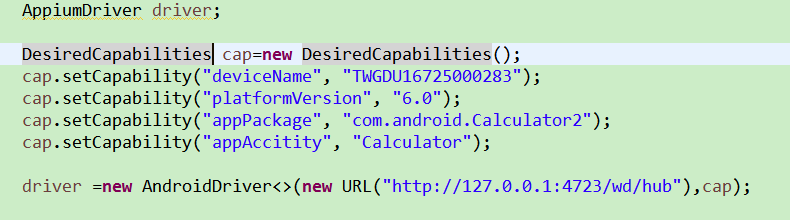
补充：

1. UISelector 是在定义UiObject的对象：UiObject object=new UiObject(s)
2. 若使用UICollection时，使用hierarchyviewer.bat查看viewGroup的名字

# 第七章 APPIUM

1. DesiredCapabilities常用参数

|  |  |
| --- | --- |
| 参数 | 用途 |
| autimationName | 指定自动化引擎，默认appium |
| platformName | 平台名称，Android |
| platformVersion | 平台版本 |
| deviceName | 测试设备名称 |
| app | 待测应用的绝对路径 |



1. DesiredCapabilities---------Android测试常用参数

|  |  |
| --- | --- |
| 参数 | 用途 |
| appActivity | 应用启动的Activity名称 |
| appPackage | 应用的包名 |
| deviceReadyTimeout | 等待设备就绪的时间 |
| useKeystore | 是否使用自定义的keystore对应用重签名 |
| unicodeKeyboard | 支持中文输入 |

<http://appium.io/slate/cn/master/?ruby#appium-服务器参数>

1. Appnium界面元素定位：

1.findElement(By)：

2.findElementByClassName();

3.findElementById

4.findElementByAccessbilityId

5.findElementByXPath

6.findElementAndroidUiautomator

1. Appnium界面等待方法：

1.thread.sleep(ms)

2.driver.manage().timeouts().implicitlyWait(Second,TimeUnit.SECONDS)

上述三四具体代码截图及注释：（详见APPTest2.java）

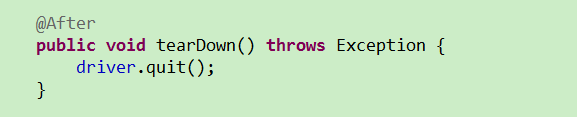


补充：

1. 在运行APPIUM时，首先运行cmd 输入appium，启动appium server,crtl+C则关闭server
2. 在appium测试中，建立project时，必须有的两个方法：

setUp():

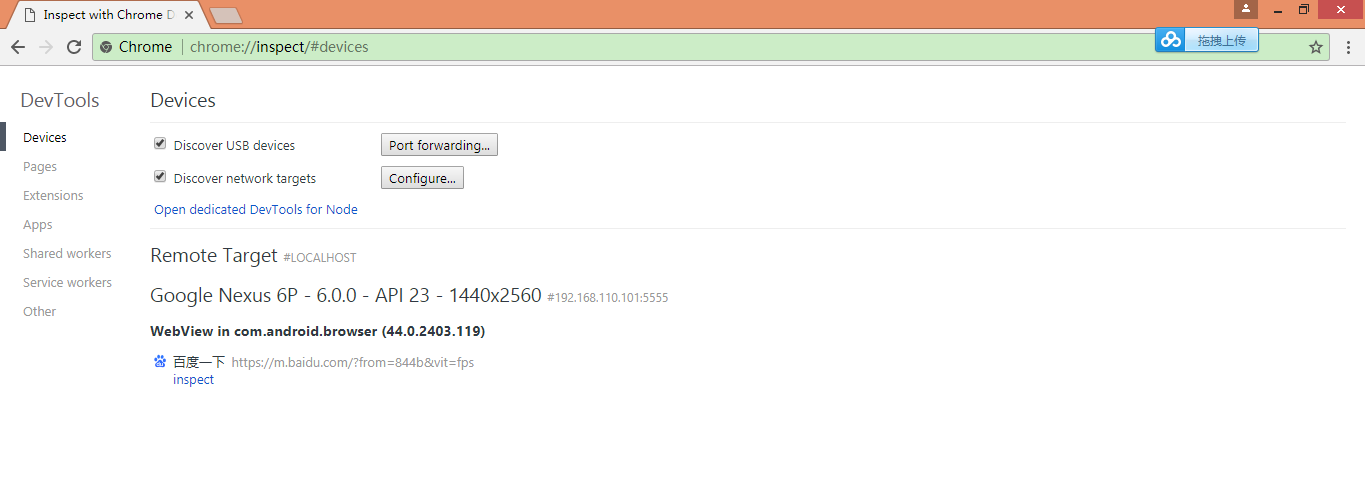




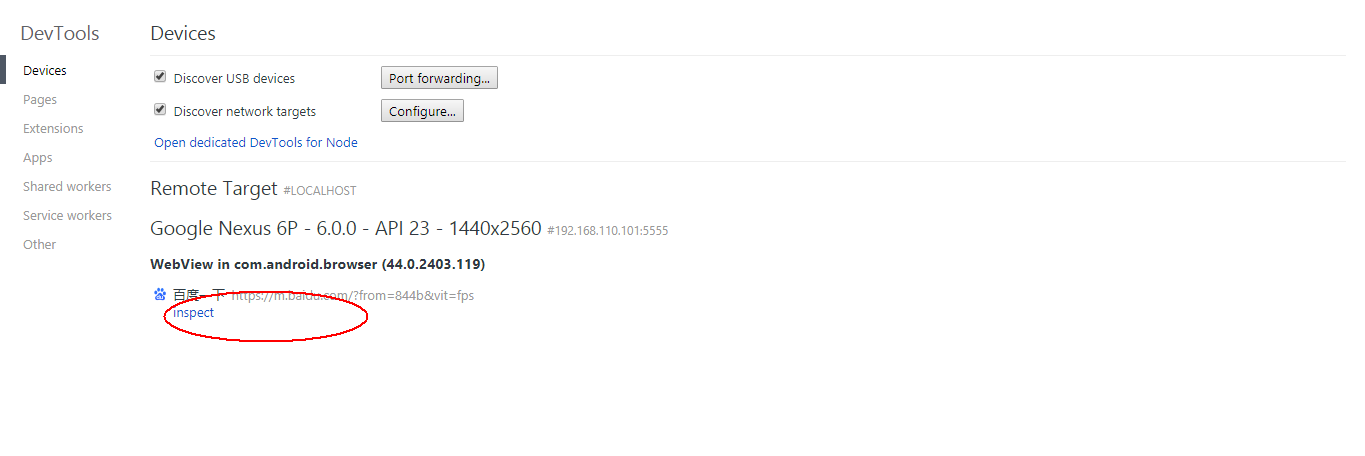
建立project时直接建立Android JUnit Test

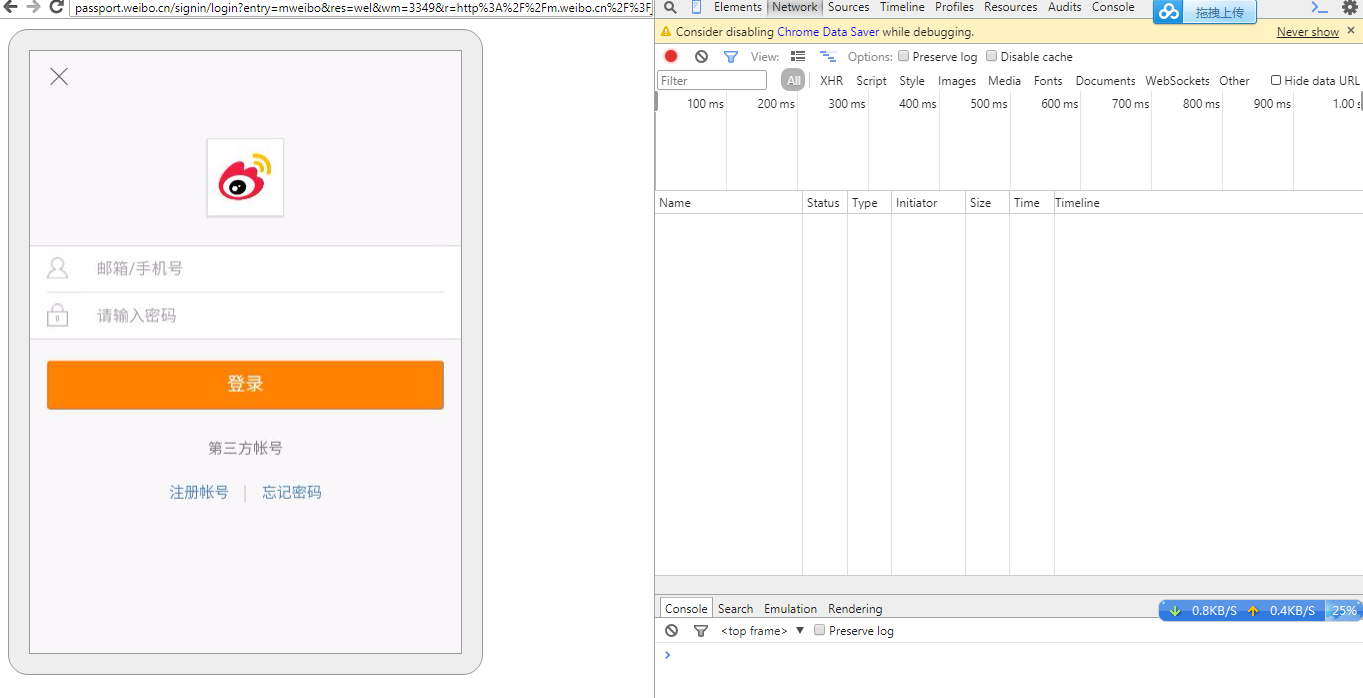
1. 测试hybrid应用时，如果是低版本的UiAutomatorviewer 则获取不到网页元素的位置，这时使用chrome的插件：

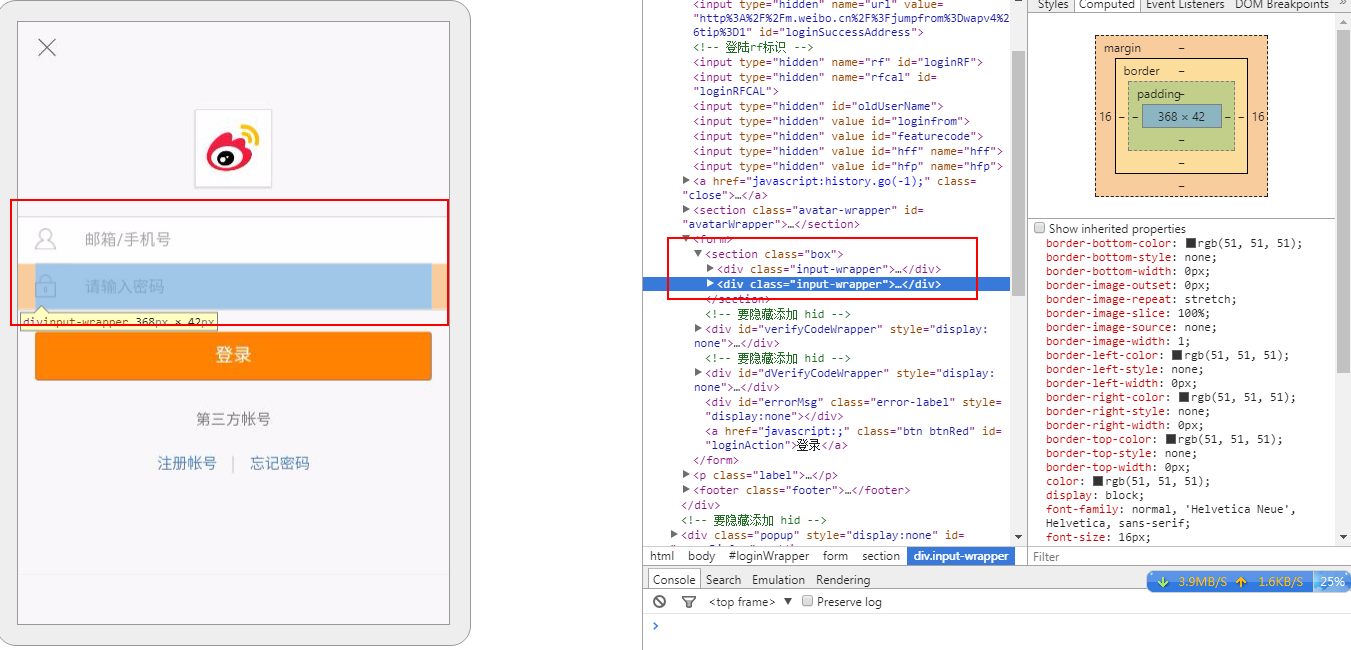
1）打开Google chrome ，输入：chrome://inspects



1. 点击inspect:



1. 
2. 找到对应输入框的基本信息：



第八章 课程总结

1. 环境准备：

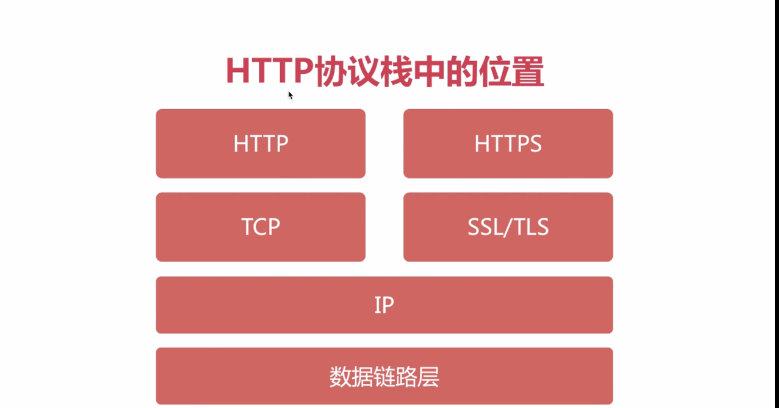
1.JAVA环境：jdk

2.android环境：Android sdk

3.ant编译环境

JMeter

1. 不同角色眼中的接口：
2. 开发眼中的接口：模块与模块之间的对接方式。interface 接口，需要具体的类来实现implements
3. 测试眼中的接口：协议接口。可以独立部署成服务的协议接口。
4. 常见的接口协议
5. HTTP:超文本传输协议
6. HTTPS:安全超文本传输协议
7. FTP：文件传输协议
8. TCP：网络控制协议
9. IP：互联网协议
10. UDP：用于数据协议
11. http协议栈的位置



1. http协议响应码

1.1XX:信息响应类，表示接收到请求并且继续处理

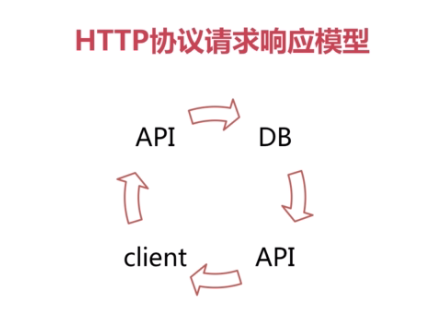
2.2XX:处理成功响应类，表示动作被成功接收、理解和接受

3.3XX:重定向响应类，为了完成指定的动作，必须接受进一步的处理

4.4XX:客户端错误，客户请求包含语法错误或者不能正确执行（404：请求资源不存在）

5.5XX:服务端错误，服务器不能正确执行一个正确的请求（服务端超时）

1. http协议响应模型



1.http协议请求响应模型（一次请求的生命周期或简单的软分层）

场景：登录

1.客户端发起请求到API接口层

1.1 用户在客户端填写用户名和密码，点击登录，发送请求

2.API接收到客户端发起的用户请求

2.1 API对业务逻辑进行验证

2.1.1 验证用户名与密码是否合法

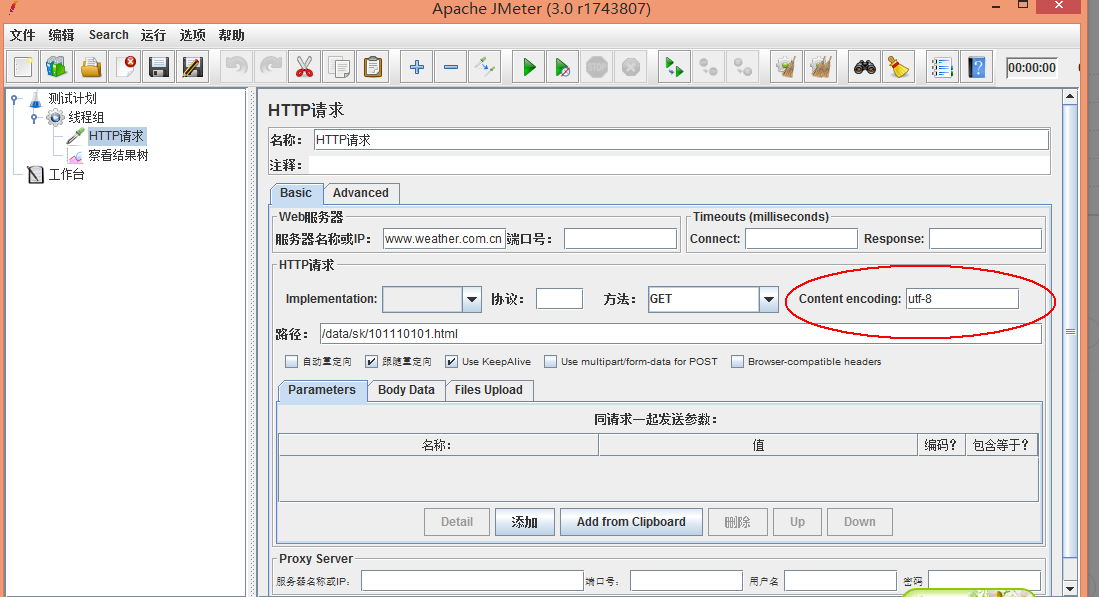
如果验证失败，既用户名或密码不合法，那么需要给客户端返回响应码。

3.API会将用户输入的数据发送给db层

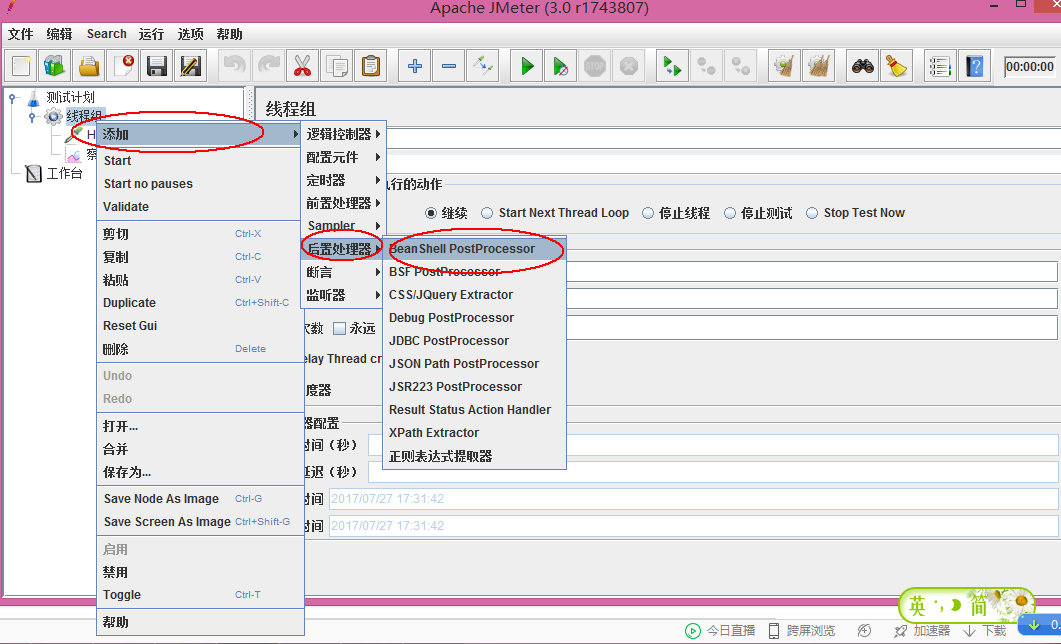
Create/Read/Upadte/Delete

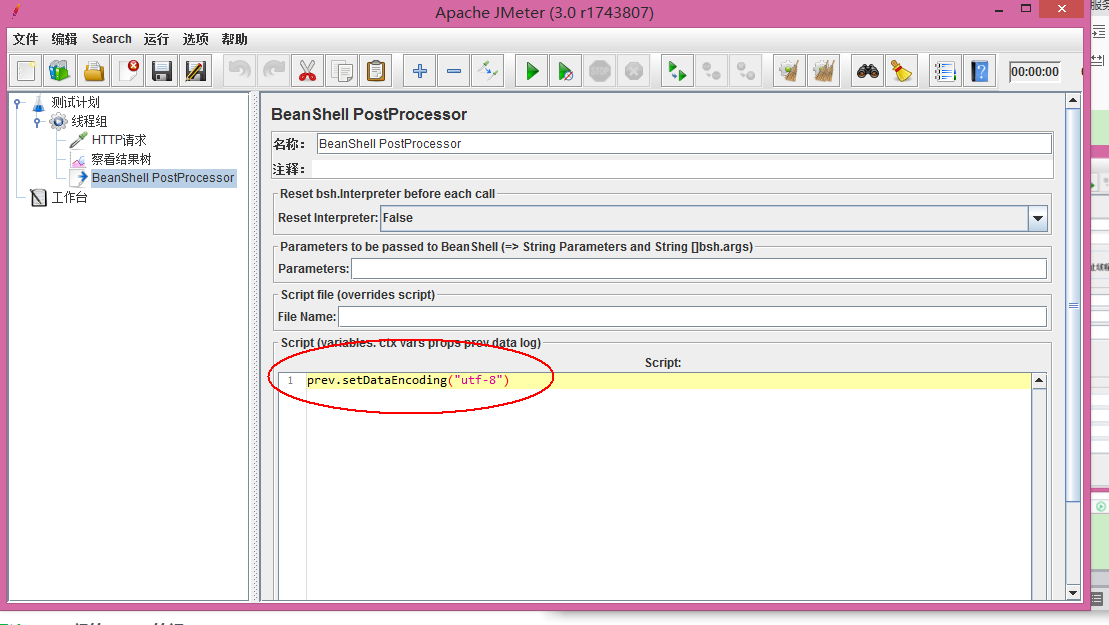
3.1 数据库查询成功则返回1，失败则返回0

1. DB会将返回的查询数据库的条目数给API
2. API返回成功或失败的状态码给客户端
3. 客户端将返回信息提示给用户
4. 客户端：功能测试、自动化测试
5. 接口层：功能测试、性能测试、自动化测试
6. 数据库层：可以将开发人员使用的sql语句单独拿出来进行性能测试
7. 常用的http协议的请求方式
8. get请求：将请求参数列出来。向特定的资源发出请求。
9. Post请求：只是展示请求地址。向指定资源提交数据进行处理请求。数据被包含在请求体中。POST请求可能会导致新的资源的建立和已有资源的修改。
10. Options:返回服务器针对特定资源所支持的HTTP方法。也可以利用向web服务器发送“”“\*”的请求来测试服务器功能
11. HEAD：向服务器索与GET请求相一致的响应，只不过响应将不会被返回。这一方法可以再不必传输整个响应内容的情况下，就可以获取包含在响应小消息头中的元信息
12. PUT：向指定资源位置上传其最新内容
13. DELETE:请求服务器删除Request-URL所标识的资源
14. TRACE:回显服务器收到的请求，主要用于测试和诊断
15. CONNECT:
16. JMeter的功能介绍
17. BS架构应用性能
18. HTTP协议接口功能与性能
19. FTP协议接口功能与性能
20. MySql数据库性能
21. MongDB数据库性能
22. 支持自定义Java组件开发
23. JMeter乱码问题

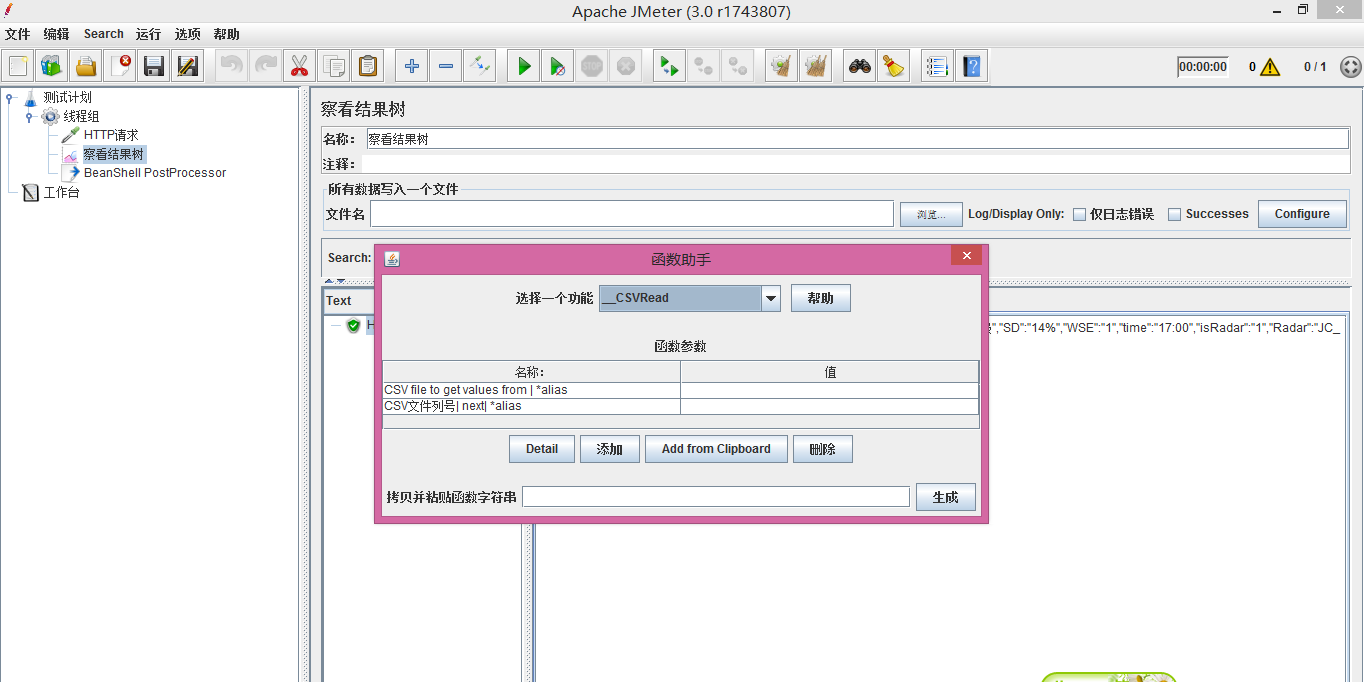
1.

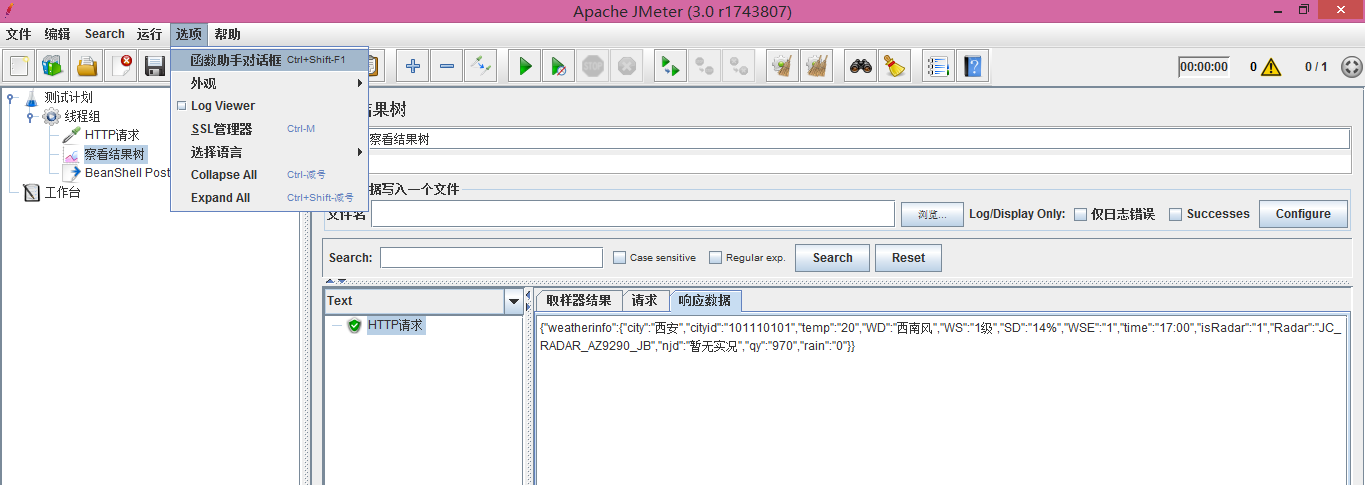
设置编码格式为utf-8

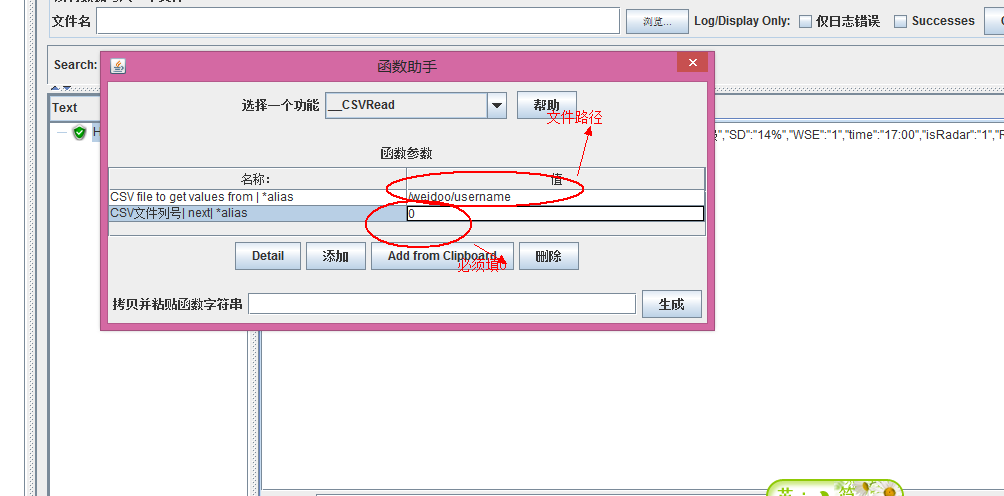
1. 修改property配置文件，找到编码行，修改
2. 添加组件：

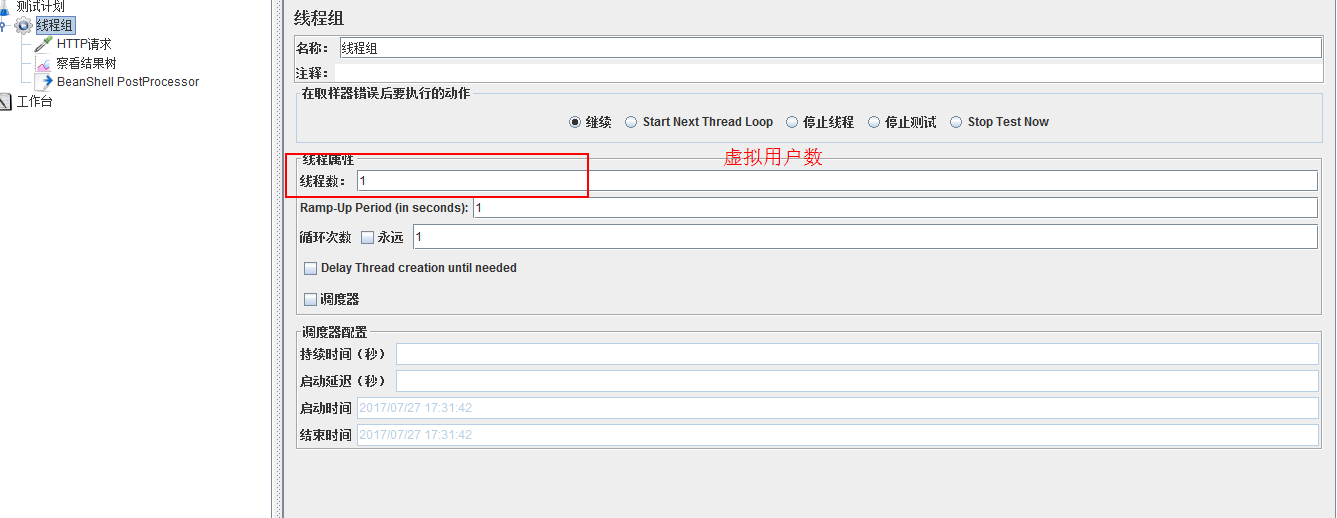


1. 参数化：







1. 
2. Get请求与post请求

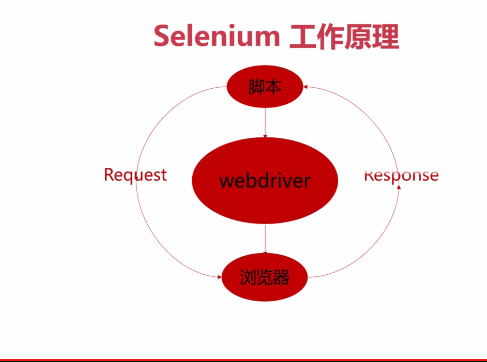
Selenuim

1. 什么是自动化测试？

1. 为什么要做自动化？
2. 减少人力成本
3. 完成大量重复性工作
4. 提高测试效率
5. 保证工作的一致性，增加信任度
6. 完成手工不能完成的工作
7. 是否适合做自动化？
8. 时间
9. 人员
10. 系统
11. 领导
12. 在什么情况下做自动化？

1.UI测试

1. 什么样的项目需要做自动化？
2. 需求变更慢
3. 周期长
4. 脚本可重复性利用
5. 自动化工具、框架的选择？
6. Selenium Webdriver
7. FireBug
8. FirePath
9. 简介
10. Selenuum RC/Selenuum IDE/Selenuum Webdriver/Selenuum Grid
11. 工作原理：



1. FireBug安装

1.火狐浏览器

数据库基本知识：

1. 创建数据库表

CREATE TABLE table\_name(

column\_name column\_type

);

实例：

CREATE TABLEIF NOT EXITS 'table\_A'(

'id' int unsigned auto\_increment,

'name' varchar(10) NOT NULL

'date' DATE

)

1. 删除数据库表

DELETE TABLE table\_name;

1. 在数据库中插入数据

insert into table\_name(column\_name1,column\_name2,column\_name3....)

values

(value1,value2,value3...)

1. 查询数据库中的数据

select \*

from table\_name

where cause

offset M

limit n

1. 查询语句可以查询一个表或者多个表，表之间使用逗号进行分割
2. select命令可以读取一条或者多条记录
3. 你可以用（\*）来代替其他字段，select会返回表中的所有字段
4. 你可使用where语句来包含任何条件
5. 可以使用offset指定SELECT语句查询数据偏移量
6. 使用limit属性来设定返回的记录数：select \* from t order by id limit 30,10

(查找表中记录31-40的记录)

1. 更新数据表

UPDATE table\_name SET column\_name1=new\_value1,column\_name2=new\_value2

where Clause

1. 可以同时更新一个或者多个字段
2. 可以在where子句中指定任何条件
3. 可以在一个单独表中同时更新数据
4. Delete语句

DELETE FROM table\_name

where Clause

1. 若没有where子句，则删除整张表
2. 可以在where子句中指定任何条件
3. 在单词表中一次性删除记录
4. LIKE语句

SELECT filed1,filed2,...filedN

FROM table\_name

where filed1 LIKE condition1 (and|or)value2="A"

1. where子句中可以使用“=”来设定获取数据的条件
2. LIKE子句中使用“%”来表示任意字符。
3. 可以在where子句中指定任何条件
4. 可以在where子句中使用like子句
5. 可以使用like子句代替=
6. Like通常与%一同使用，类似于一个元字符的搜索
7. 使用and或者or指定一个或多个条件
8. Union操作字符

SELECT expression1, expression2, ... expression\_n

FROM tables

[WHERE conditions]

UNION [ALL | DISTINCT]

SELECT expression1, expression2, ... expression\_n

FROM tables

[WHERE conditions];

1. 用于连接两个以上的select语句的结果组合到一个结果集中，去除重复数据
2. Expression1.为要检索的列
3. Tables ：要检索的数据表
4. WHERE conditions :可选，检索条件
5. Distinct：可选，删除结果集中重复的数据、默认情况下，union已经去除重复数据
6. All：可选，返回所有结果集，包含重复数据
7. 排序

SELECT field1, field2,...fieldN table\_name1, table\_name2...

ORDER BY field1, [field2...] [ASC [DESC]]

1. 可以使用任何字段来作为排序的条件，从而返回排序后的查询结果
2. 可以设定多个字段来排序
3. 可以使用asc或者desc关键字来设置查询结果按照升序或者降序排列
4. 可以使用where like来设置条件
5. GROUP BY语句

SELECT column\_name, function(column\_name)

FROM table\_name

WHERE column\_name operator value

GROUP BY column\_name;

1. group by语句根据一个或者多个列对结果集进行分组
2. 在分组的列上我们可以使用COUNT、sum，avg等函数
3. 事务
4. 事务的定义
5. 事务的四个条件：
6. 原子性：一组事务，要么成功，要么撤回
7. 稳定性：有非法数据（外键约束之类），事务撤回
8. 隔离性：事务独立运行，一个事务处理后的后果，影响了其他事务，那么其他事务会撤回，事务的100%隔离，需要牺牲速度。
9. 可靠性：软、硬件崩溃后，INNODB数据表驱动会利用日志文件重构修改。
10. 事务的控制语句
11. begin或start transaction：显式的开启一个事务
12. Commit：提交事务，并使已对数据库进行的所有修改称为永久性的
13. Rollback:回滚会结束用户的事务，并撤销正在进行的所有未提交的修改
14. SAVEPOINT identifier：SAVEPOINT 允许在事务中创建一个保存点，一个事务中可以有多个SAVEPOINT
15. RELEASE SAVEPOINT identifier：删除一个事务的保存点，当没有指定的保存点时，执行语句会抛出一个异常
16. ROLLBACK TO identifier：把事务回滚到标记点
17. SET TRANSCATION：用来设置事务的隔离级别
18. 事务处理方式
19. 用begin、rollback、commit来实现

1.begin开始一个事务

2.rollback事务回滚

3.commit事务确认

1. 直接用set来改变MySQL的自动提交模式

1.SET AUTOCOMMIT=0禁止自动提交

2.SET AUTOCOMMIT=1 开始自动提交

1. 防止sql注入
2. 永远不要相信用户的输入。对用户的输入进行校验，可以通过正则表达式，或限制长度；对单引号和双“-”进行转换
3. 永远不要使用动态拼装aql，可以使用参数化的sql或者直接使用存储过程进行数据查询存取
4. 永远不要使用管理员的权限的数据库连接，为每个应用使用单独的权限有限的数据库连接。
5. 不要把机密信息直接存放，加密或者hash掉密码或者敏感的信息
6. 应用的异常信息应该给出尽可能少的提示，最好使用自定义的错误信息对原始错误信息进行包装
7. sql注入的检测方法一般采取辅助软件或网站平台来检测，软件一般采用sql注入检测工具isky.
8. 存储过程：

create proc |procedure procedure\_name

[@参数数据类型 =默认值 output

]

As

SQL\_STATEMENTS

GO

存储过程与SQL语句对比

优点：

1. 提高性能：SQL语句在创建过程时进行分析和编译。存储过程是预编译的，在首次运行一个存储过程时，查询优化器对其进行分析优化，并给出最终被存在系统表中的存储计划，这样，在执行过程中便可节省此开销。
2. 降低网络开销：存储过程调用时只需用提供存储过程名和必要的参数信息，从而可减低网络的流量
3. 便于代码移植：数据库专业人员可以随时对存储过程进行修改，但对应用程序却毫无影响，从而极大的提高了程序的可移植性
4. 更强的安全性：
5. 系统管理员可以执行的某一个存储过程进行权限限制，避免非授权用户对数据 的访问
6. 在通过网络调用过程时，只有对执行过程的调用是可见的。因此，恶意用户无法看到表和数据库对象名称，嵌入自己的SQL语句或者搜索关键字
7. 使用过程参数有助于避免SQL注入攻击。因为参数输入被视作文字值而非可执行代码，所以，攻击者将命令插入过程中的SQL语句并损坏安全性将更为困难
8. 可以对过程进行加密，这有助于对源代码进行模糊处理

缺点：

1. 存储过程需要专门的数据库开发人员进行维护，但实际情况是，往往由程序开发人员兼职
2. 设计逻辑变更，修改存储过程没有SQL灵活
3. 函数