


浅谈自动化测试工具 Appium

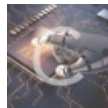
转载

慧乐课堂软件测试  于 2020-11-02 19:20:18 发布  125  收藏

分类专栏: [appium](#) [软件测试](#) [自动化测试](#)

原文链接: <https://testerhome.com/topics/11318>

版权



[appium](#) 同时被 3 个专栏收录

2 篇文章 0 订阅

订阅专栏



[软件测试](#)

149 篇文章 10 订阅

订阅专栏



[自动化测试](#)

4 篇文章 0 订阅

订阅专栏

一、简单介绍

(一) 测试对象

appium 是一个自动化测试开源工具，支持iOS和 Android平台上的原生应用，web应用和混合应用。

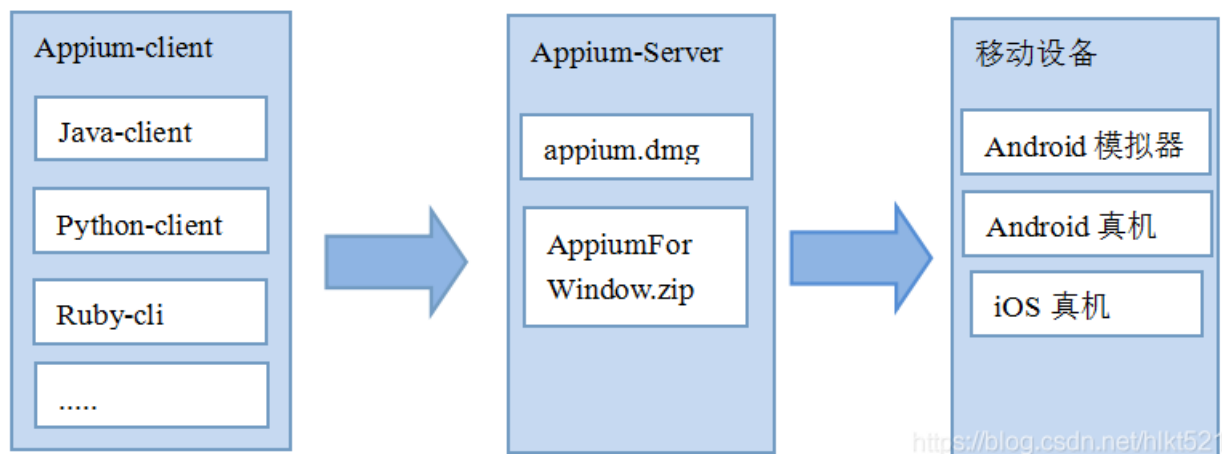
- 移动原生应用：单纯用ios或者android开发语言编写的、针对具体某类移动设备、可直接被安装到设备里的应用，一般可通过应用商店获取，比如某个游戏app；
- 移动web应用：使用移动浏览器访问的应用（appium支持iOS上的Safari和Android上的Chrome），不需要下载到设备上，而是通过浏览器直接访问，比如H5九曲封神游戏；
- 应用：同时使用网页语言与程序语言开发，通过应用商店区分移动操作系统分发，用户需要安装使用的移动应用，比如手机上的淘宝客户端

(二) 支持平台及语言

appium是跨平台的，支持OSX，Windows以及Linux系统。它允许测试人员在不同的平台（iOS，Android）使用同一套API来写自动化测试脚本，这样大大增加了iOS和Android测试套件间代码的复用性

appium支持多语言，采用C/S设计模式，只要满足client能够发送http请求给server即可

(三) 工作原理



如图所示，appium主要分为三个点：client端，server端，及测试移动设备

- client端：用各类语言写的测试脚本
- erver端：我们安装appium工具的那部分，专门用来监听并接收来自client端的请求，转发请求并控制移动设备执行测试，默认开启并监听4723
- 接口移动设备：支持三类设备，我们的最终目的：在设备上自动执行指定的操作

整个过程：

第一，在某台服务器上安装appium工具（server端），开启进程（默认4723接口）进行监听；

第二，编写测试脚本，放置到服务器上，执行该测试脚本（相当于发送命令给server端）

第三，移动设备接收到server端发送过来的命令，执行指定的操作

(四) 安装工具

- 测试语言，如python
- appium客户端
- appium服务端
- 移动设备，若使用虚拟机则需进行安装

二、环境搭建

(一) 安装Android SDK

1、Android SDK（Software Development Kit，软件开发工具包）提供了 Android API 库和开发工具构建，测试和调试应用程序，可以看做用于开发和运行Android应用的一个软件

2、提供小工具，比如adb、aapt、uiautomatorview

3、测试设备使用安卓模拟器，这一步决不能跳过（

二) 安装 appium Server

1、下载地址：<https://github.com/appium/appium-desktop/releases> 2、下载appium安装包（appium-desktop-Setup-1.2.7.exe，appium-desktop-1.2.7.dmg等）

3、安装，并配置appium环境变量

（三）安装 python-client

- 1、先安装编程语言，比如python语言
- 2、安装Appium-Client，python的话可使用pip安装：pip install Appium-Python-Client

三、应用及操作

（一）调用appium过程

- 1、配置手机设备参数，告诉server端我想调起的是哪一台手机设备
- 2、抓取手机上应用的控件，指定对应的控件来进行操作
- 3、对抓取到的控件进行操作，比如点击、填写参数等

（二）第一步，配置手机设备参数

```
#Android environment
import unittest
from appium import webdriver
desired_caps = {}
desired_caps['platformName'] = 'Android'
desired_caps['platformVersion'] = '4.2'
desired_caps['deviceName'] = 'Android Emulator'
desired_caps['app'] = PATH('../..../apps/selendroid-test-app.apk')
desired_caps['appPackage'] = package
desired_caps['appActivity'] = activity

self.driver = webdriver.Remote('http://localhost:4723/wd/hub', desired_caps)
```

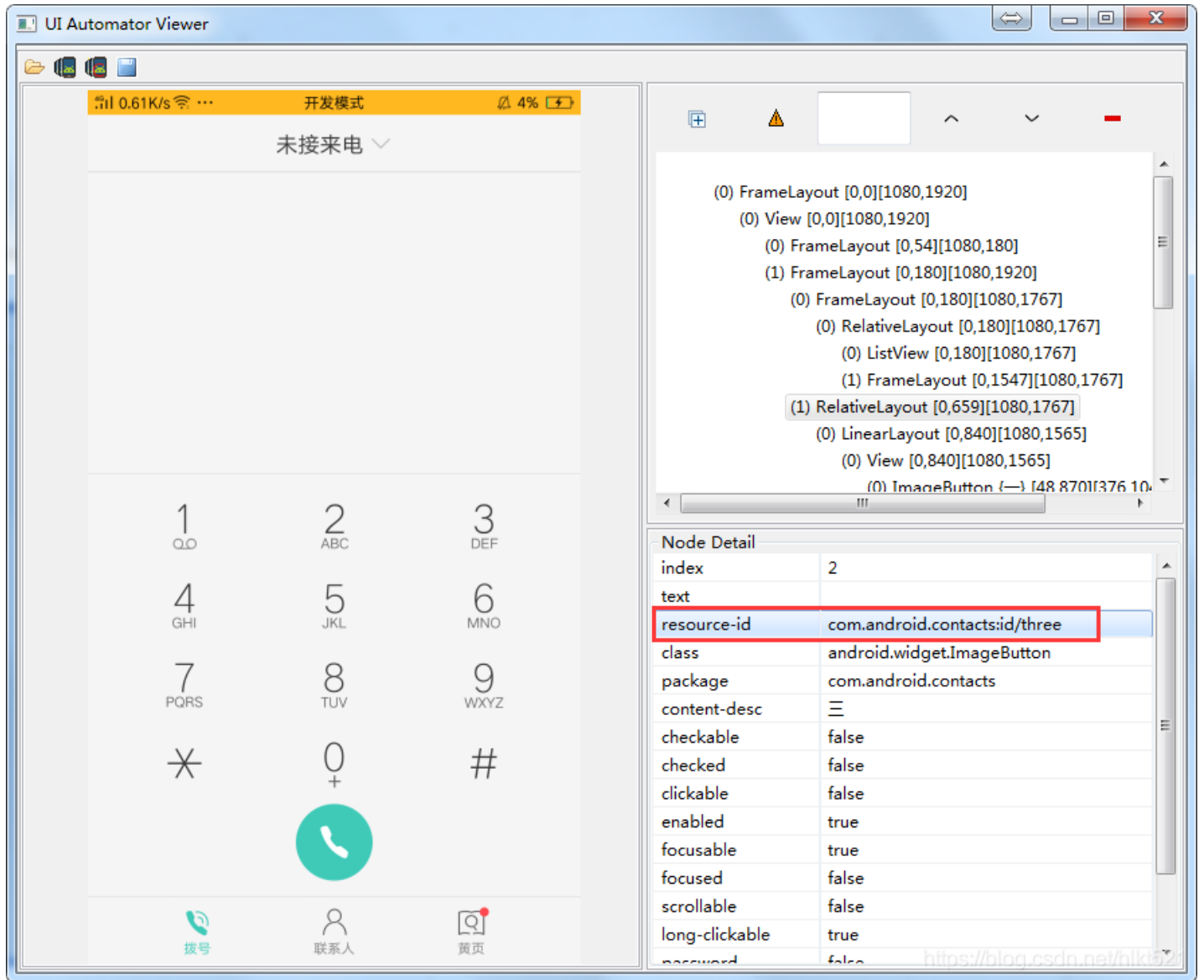
常见参数解释：

```
deviceName: 指定启动设备，比如Android Emulator、iPhone Simulator等
automationName: 指定自动化引擎，默认appium
platformName: 指定移动平台，Android或者iOS
platformVersion: 指定平台的系统版本。例如指定Android系统版本为4.2
appActivity: 待测试app的Activity，注意，原生app的话要在activity前加个“.”
appPackage: 待测试app的包名（package）信息
```

（三）第二步，抓取手机上应用的控件

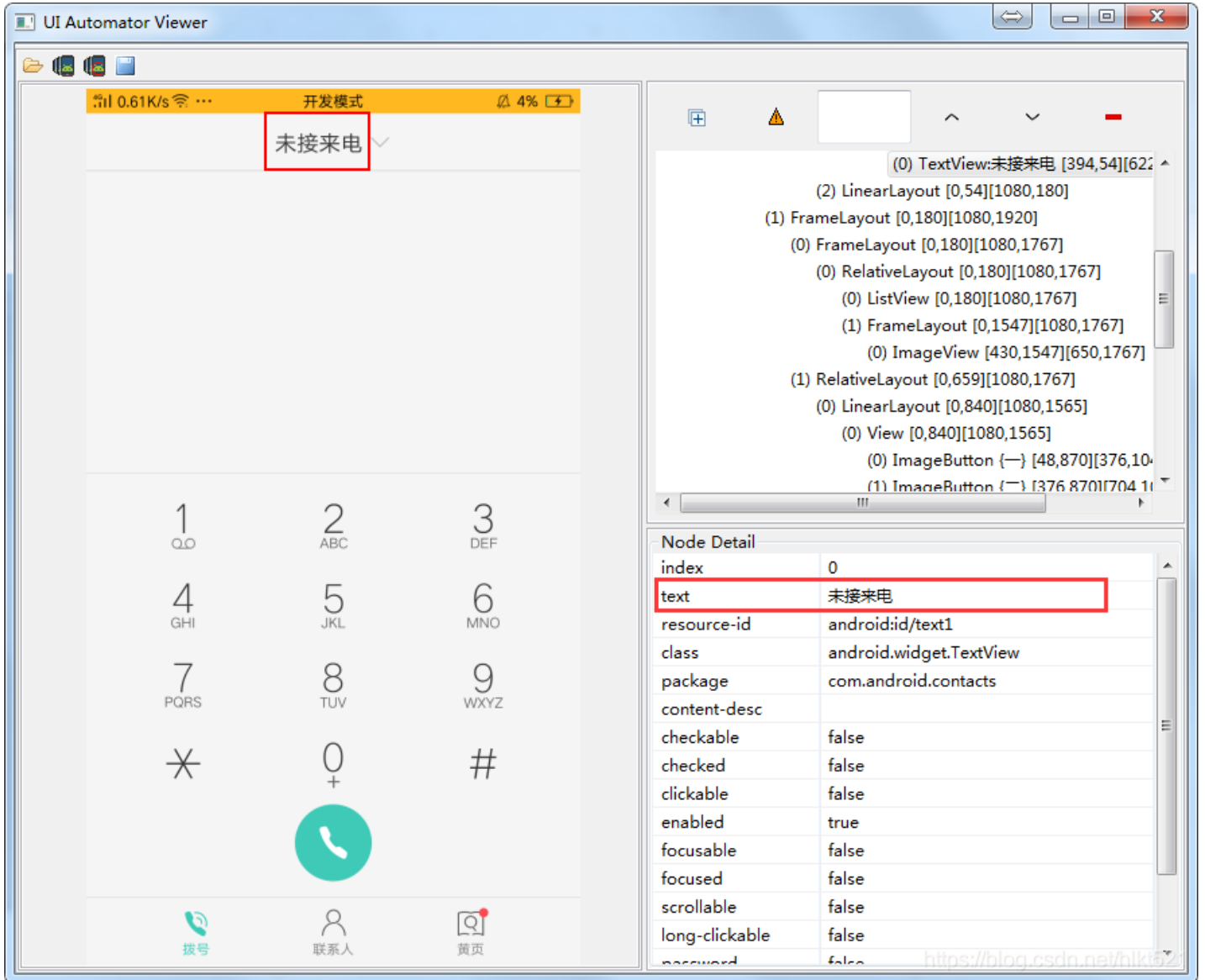
通过Android SDK内置工具uiautomatorviewer.bat来查看手机应用的控件参数（该工具位于 /tools/bin/ 目录下）

1、id定位



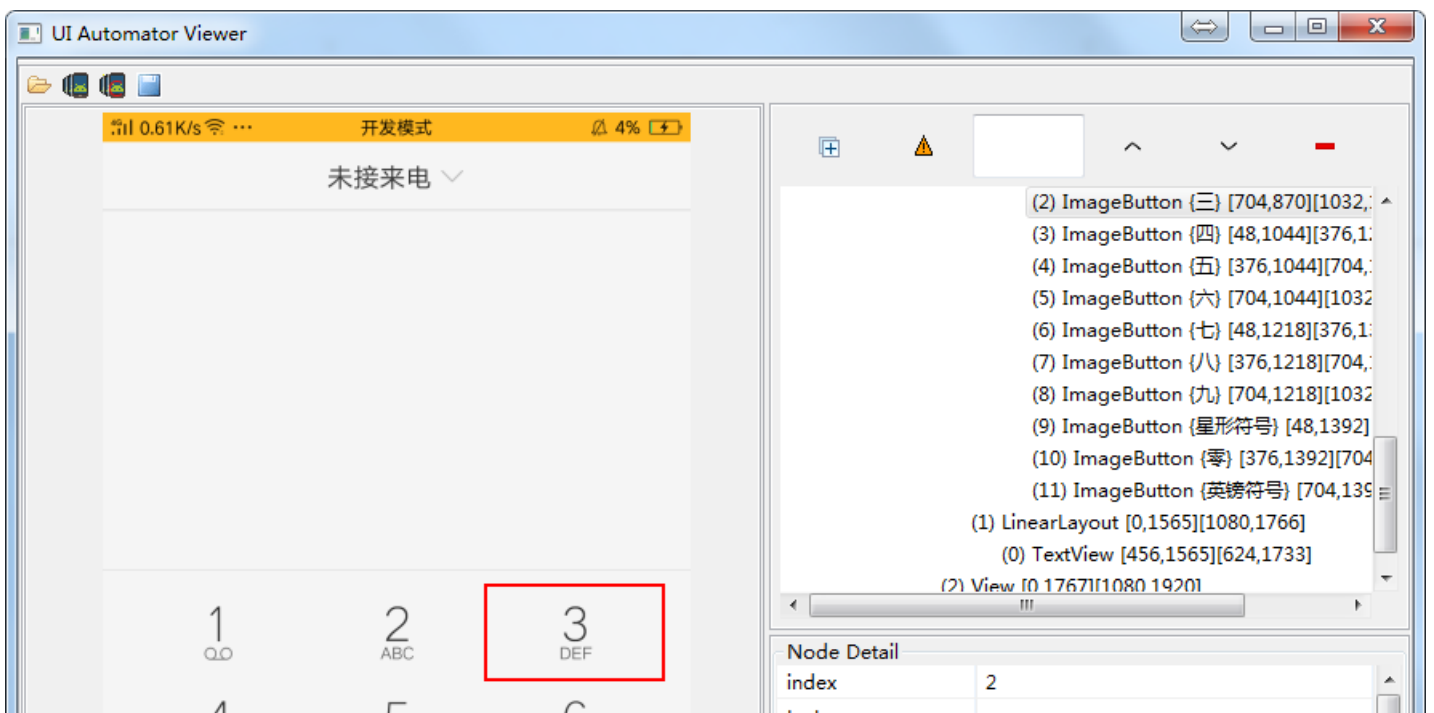
使用方法： `driver.find_element_by_id('com.android.contacts:id/three')`

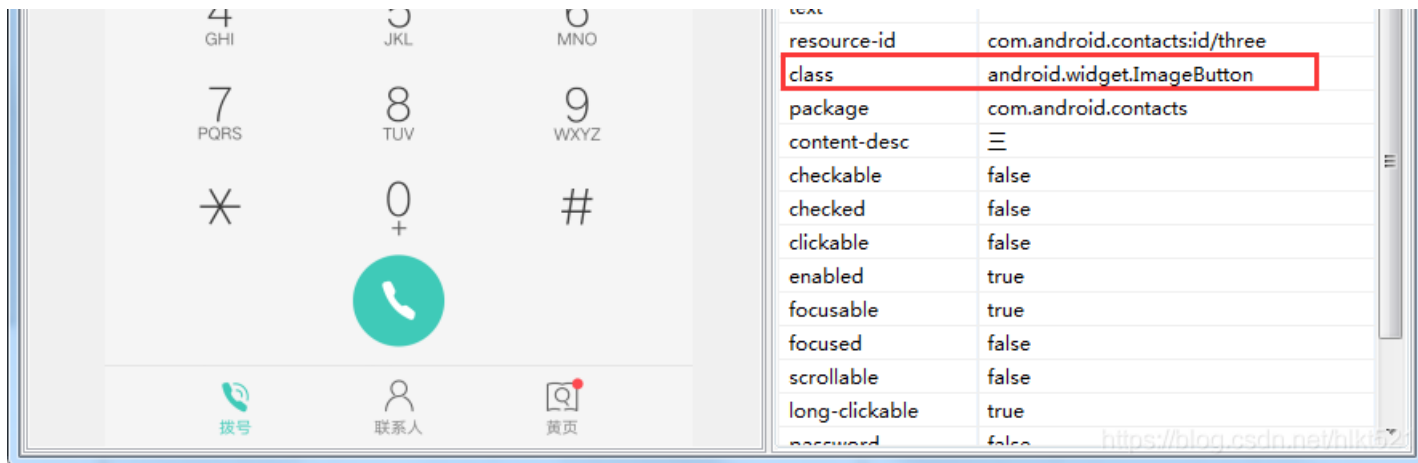
2、name定位



使用方法：`el = self.driver.find_element_by_name('未接来电')` `el = self.driver.find_elements_by_name('未接来电')`

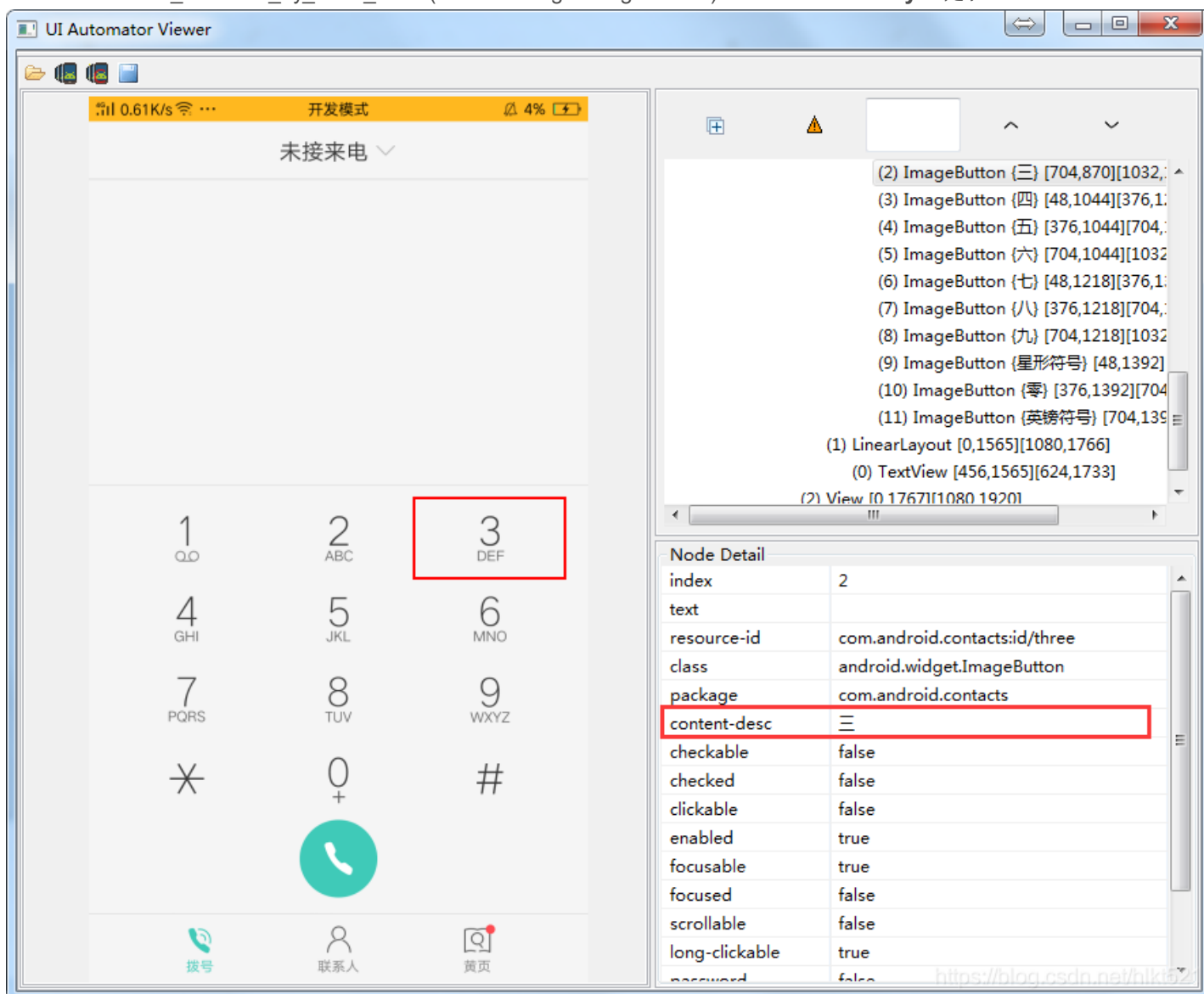
3、class name 定位





使用方法: `els=self.driver.find_element_by_class_name('android.widget.ImageButton')`

`els=self.driver.find_elements_by_class_name('android.widget.ImageButton')` 4、4、Accessibility ID定位

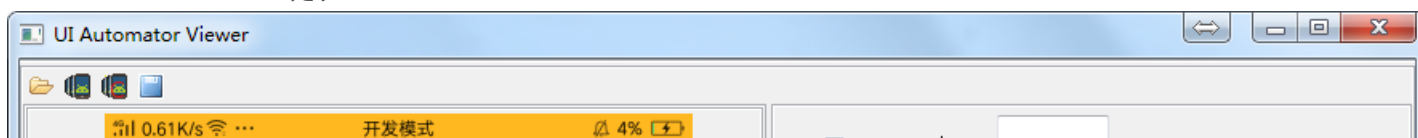


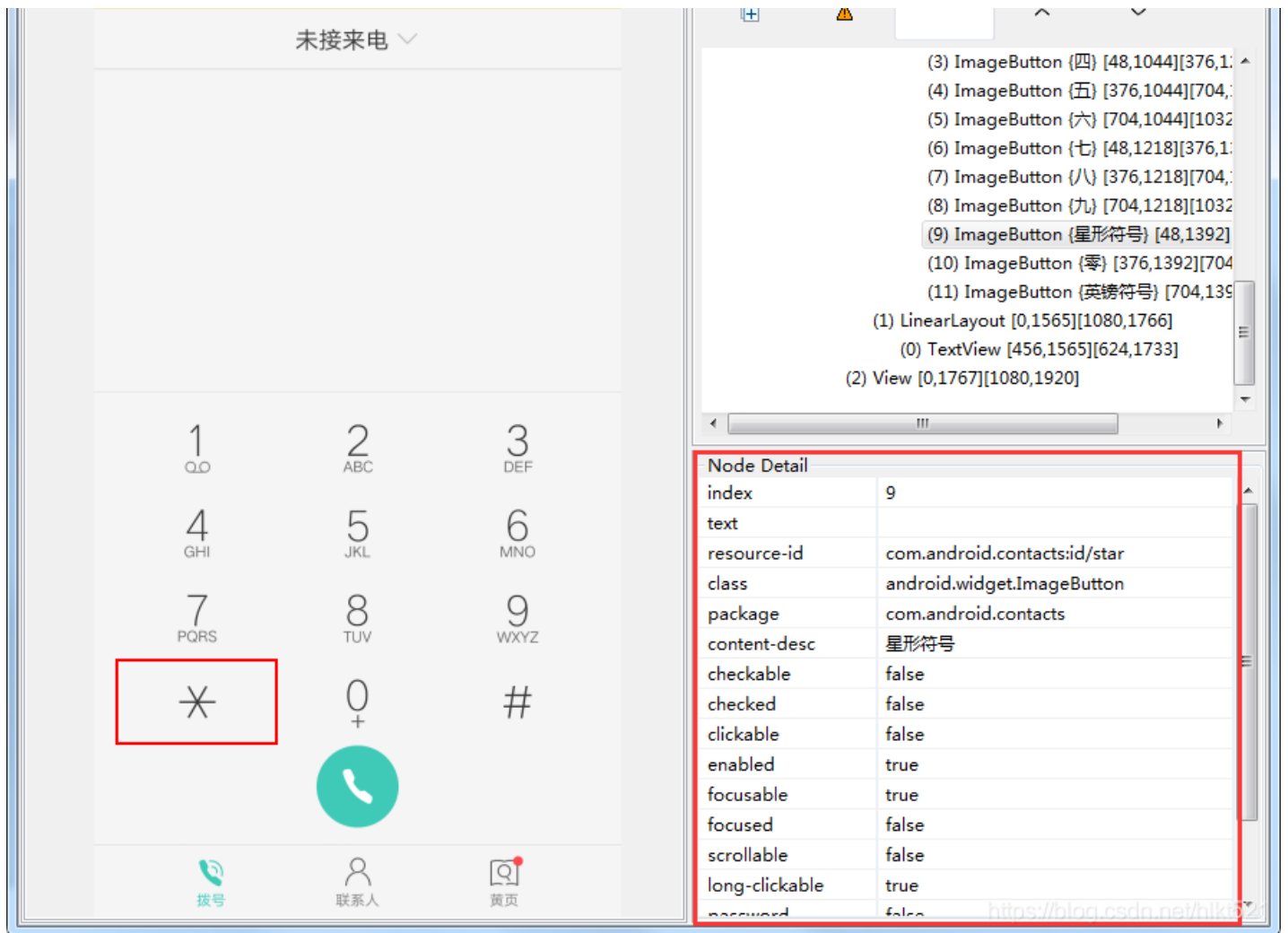
使用方法:

`el = self.driver.find_element_by_accessibility_id('☰')`

`el = self.driver.find_elements_by_accessibility_id('☰')`

5、android uiautomator定位





使用方法:

```
els = self.driver.find_element_by_android_uiautomator('new UiSelector().description(星形符号)')
```

```
els =self.driver.find_elements_by_android_uiautomator('new UiSelector().clickable(false)')
```

(四) 第三步，操作控件

1.scroll

scroll(self, origin_el, destination_el):

从元素origin_el滚动至元素destination_el

举例: driver.scroll(e1, e2)

用法: driver.scroll(e1,e2)

2.tap

tap(self, positions, duration=None):

模拟手指点击（最多五个手指），可设置按住时间长度（毫秒）

举例: driver.tap([(100, 20), (100, 60), (100, 100)], 500)

用法: driver.tap([(x,y),(x1,y1)],500)

3. swipe

swipe(self, start_x, start_y, end_x, end_y, duration=None):

从A点滑动至B点，滑动时间为毫秒

举例: driver.swipe(100, 100, 100, 400)

用法: driver.swipe(x1,y1,x2,y2,500)

4、keyevent

keyevent(self, keycode, metastate=None):

发送按键码（安卓仅有），按键码可以上网址中找到

用法: driver.keyevent('4')

5、press_keycode

press_keycode(self, keycode, metastate=None):

发送按键码（安卓仅有），按键码可以上网址中找到

用法: driver.press_keycode('4')

6.text

text(self):

返回元素的文本值

用法: element.text

7.click

click(self):

点击元素

用法: element.click()

8.get_attribute

get_attribute(self, name):

获取某元素的相关值

用法: element.get_attribute("name")

9.size

size(self):

获取元素的大小（高和宽）

用法 driver.element.size

10、page_source

page_source(self):

获取当前页面的源

用法: driver.page_source

11.quit

quit(self):

退出脚本运行并关闭每个相关的窗口连接

举例: driver.quit()

四、总结

1、Appium跨平台，支持Android和IOS自动化测试。

2、Appium支持多语言，如java、Object-C、JavaScript、Php、Python、Ruby、C#、Clojure等等

3、支持原生应用，web应用和混合应用

4、仅仅支持UI测试