

Please visit website: <http://cxyroad.com>

3种方式自动化控制APP

=====

自动化控制APP不管是在工作还是生活方面，都可以帮助我们高效地完成任
务，节省时间和精力。本文主要介绍自动化控制APP的3种常用方式。

1、Python + adb

> 这种方式需要对Android有一些基本的了解。adb是一种用于调试Android应
用程序的工具。使用Python和adb可以轻松实现自动化控制移动端APP。

1.1、特点

这种方式最简单，但是控制效果也最粗糙。不同的手机对应的元素x,y轴的位置
不同，所以不适合操作不同尺寸的所有手机。这种方式也只适合于开发者，对
于普通用户使用成本过高。

1.2、使用步骤

1、安装Android SDK

具体细节略过，自行google安装。安装完毕后，配置好`ANDROID_HOME`环
境变量。

2、安装Python

具体细节略过，自行google安装。

3、打开手机的开发者模式

同时开启`USB调试`和`显示指针位置`。

4、此时操作手机

可以看到有2根轴，同时最上方会显示页面焦点元素的`x,y轴`位置。

```
!(https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/1c9e2477550440c1a69d912484c212c3~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.aawebp#?w=634&h=1360&s=481541&e=png&b=fbf8f8)
```

5、使用Python代码+adb简单控制APP

...

```
import time
import subprocess
```

```
# 点击某个位置
```

```
subprocess.getoutput(f"adb -s Y2J7N17C27000069 shell input tap 564 1861")
```

```
time.sleep(2)
```

```
subprocess.getoutput(f"adb -s Y2J7N17C27000069 shell input tap 188 980")
```

```
subprocess.getoutput(f"adb -s Y2J7N17C27000069 shell input tap 869 1808")
```

```
time.sleep(4)
```

```
subprocess.getoutput(f"adb -s Y2J7N17C27000069 shell input tap 320 965")
```

```
# 输入数据
```

```
subprocess.getoutput(f"adb -s Y2J7N17C27000069 shell input text 15850501595")
```

```
# 按返回键
```

```
subprocess.getoutput(f"adb -s Y2J7N17C27000069 shell input keyevent KEYCODE_BACK")
```

```
subprocess.getoutput(f"adb -s Y2J7N17C27000069 shell input tap 512 1120")
```

```
# 输入数据
```

```
subprocess.getoutput(f"adb -s Y2J7N17C27000069 shell input text 15850501595")
```

```
subprocess.getoutput(f"adb -s Y2J7N17C27000069 shell input keyevent KEYCODE_BACK")
```

```
subprocess.getoutput(f"adb -s Y2J7N17C27000069 shell input tap 843 1824")
```

...

6、adb常用命令

...

- 查看手机设备：adb devices
- 查看设备型号：adb shell getprop ro.product.model
- 查看电池信息：adb shell dumpsys battery
- 查看设备ID：adb shell settings get secure android_id
- 查看设备IMEI：adb shell dumpsys iphonesubinfo
- 查看Android版本：adb shell getprop ro.build.version.release
- 查看手机网络信息：adb shell ifconfig
- 查看设备日志：adb logcat
- 重启手机设备：adb reboot
- 安装一个apk：adb install /path/demo.apk
- 卸载一个apk：adb uninstall <package>
- 查看系统运行进程：adb shell ps
- 查看系统磁盘情况：adb shell ls /path/
- 手机设备截屏：adb shell screencap -p /sdcard/aa.png
- 手机文件下载到电脑：adb pull /sdcard/aa.png ./
- 电脑文件上传到手机：adb push aa.png /data/local/
- 手机设备录像：adb shell screenrecord /sdcard/ab.mp4
- 手机屏幕分辨率：adb shell wm size
- 手机屏幕密度：adb shell wm density
- 手机屏幕点击：adb -s xxxxxxxxxxxxxxxxxxxxxxxxxxxx shell input tap xvalue yvalue
- 手机屏幕滑动：adb shell input swipe 1000 1500 200 200
- 手机屏幕带时间滑动：adb shell input swipe 1000 1500 0 0 1000
- 手机文本输入：adb shell input text xxxxx
- 手机键盘事件：adb shell input keyevent xxxx

...

2、Android无障碍

> 这种方式是使用Android无障碍功能实现自动控制APP的效果。需要开启Android无障碍功能，然后编写Android代码来控制另外的APP应用。

2.1、特点

这种方式需要开发者对Android有一些开发经验。优点是：可以用Android开发出独立的apk安装包，安装到普通用户手机里，方便用户使用。

2.2、使用步骤

1、安装Android SDK、安装Android Studio

具体细节略过，自行google安装。安装完毕后，配置好`ANDROID_HOME`环境变量。

2、使用Android自带的tool工具

Android在`level-21`和之前的低版本，安装完毕后有一个`tool`工具包，但是高版本移除了此工具包。所以在安装SDK时还需要加上`level-21`版本。

使用Android自带的`tool`工具，主要是为了查看APP的页面布局 and 元素。但是`monitor`已经不可用了，只能使用`uiAutomatorViewer`。

3、利用`uiAutomatorViewer`**工具找到元素信息**

用`uiAutomatorViewer`查看页面元素所在的`x,y轴`的布局。然后编写Android代码控制点击等效果。此工具显示出来的界面如下：

4、代码示例

在`AndroidManifest.xml`代码里配置无障碍service，然后实现

`AccessibilityService`类，实现`onAccessibilityEvent`方法。后续手机界面如果有变动，内部会自动触发调用`onAccessibilityEvent`方法。

```
...
public class XXXXXAccessibilityService extends AccessibilityService {
    @Override
    public void onInterrupt() {

    }

    @Override
    public void onAccessibilityEvent(AccessibilityEvent event) {
        Log.e("无障碍", "来了");

        // 创建线程去执行任务
        new Thread(new Runnable() {
            @Override
            public void run() {
                // 后续代码
            }
        }).start();
    }
}
...
```

无障碍功能本身也提供了多种寻找页面元素的方法，比如：

```
...
/**
 * 根据ID找元素
 */
private AccessibilityNodeInfo findNodeById(String id) {
    AccessibilityNodeInfo root = getRootInActiveWindow();
    if (root == null) {
        return null;
    }
    List<AccessibilityNodeInfo> nodeList =
    root.findAccessibilityNodeInfosByViewId(id);
    if (nodeList != null) {
        for (int i = 0; i < nodeList.size(); i++) {
            AccessibilityNodeInfo node = nodeList.get(i);
            if (node != null) {
                return node;
            }
        }
    }
}
```

```
    }  
  }  
}  
return null;  
}
```

...

3、Python + Appium

> Python加Appium可以组合成为一种自动化测试工具，可以用于测试和自动化控制移动端APP。

3.1、特点

这种方式可以自动化操作APP，但是使用者基本是开发者，普通用户很难完成这一系列的操作。但是也有个优点，有些页面标记为不可点击的元素，通过这种方式是可以点击的。

Appium提供了更多的寻找页面元素的方式，比如uiautomator、xpath、id等。

3.2、使用步骤

1、安装Python

此处略过步骤，自行google。

2、安装Appium

老版本的Appium直接包含了`Appium server`和`Appium inspector`，新版本的Appium安装完毕后，需要在单独安装`Appium inspector`。通过`Appium inspector`可以查看手机当前页面的xml布局。

...

```
# 安装appium
```

```
npm i --location=global appium
```

```
# 安装自动查看UI页面的驱动  
appium uiautomator2driver
```

```
# 安装inspector  
下载安装地址: https://github.com/appium/appium-inspector?tab=readme-ov-file
```

```
# 启动server  
appium server --use-driver=uiautomator2
```

```
# 安装客户端, 代码里会使用 appium-python-client 与server通信,  
# 然后server在将指令下发到手机里的appium端  
pip3 install appium-python-client
```

...

3、使用过程

安装完毕后, 就可以通过Python代码控制App了。

- * 启动Appium-server, `appium uiautomator2driver`。
- * 启动Appium inspector, 配置好手机, 然后点击`start session`, 界面如下:

- * 通过Appium inspector查看页面布局 and 元素, 找出目标元素。

- * 编写代码

...

```
device_app_info = AppiumOptions()  
# 操作系统  
device_app_info.set_capability('platformName', 'Android')
```

```
# 操作系统版本
# device_app_info.set_capability('platformVersion', '10')
device_app_info.set_capability('platformVersion', '9')
# 设备名称
# device_app_info.set_capability('deviceName', '46F4C19402000952')
device_app_info.set_capability('deviceName', 'Y2J7N17C27000069')
# app package
device_app_info.set_capability('appPackage', 'cn.damai')
# app activity name
device_app_info.set_capability('appActivity',
'.launcher.splash.SplashMainActivity')
# 使用uiautomator2驱动
device_app_info.set_capability('automationName', 'UiAutomator2')

# 连接appium server, server地址查看appium启动信息
driver = webdriver.Remote('http://127.0.0.1:4723',
options=device_app_info)

# 找到元素, 控制元素
buy_btn = driver.find_element(AppiumBy.ANDROID_UIAUTOMATOR, 'new
UiSelector().resourceId("cn.damai:id/trade_project_detail_purchase_statu
s_bar_container_fl")')
if buy_btn:
    buy_btn.click()

...
```

4、总结

- * 以上3种方式都可以用于自动化控制移动端APP，但是又各有适用的场景。读者需根据实际情况选择其中的一种方式来实现自动化控制。
- * 以上的方式也可以认为是一种爬虫。还有一些自动化的方式是：先`分析api请求`，然后`逆向分析js`或者`逆向分析apk包`，破解其中的`加密方式`。然后直接调用api，这种方式后面单独讲解。

****本篇完结！欢迎点赞 收藏！！！****

****原文链接：**** [mp.weixin.qq.com/s/KVylEo2ri...](http://cxyroad.com/"https://mp.weixin.qq.com/s/KVylEo2riRL8yWHgPloBA")、
[www.mangod.top/articles/20...](http://cxyroad.com/"http://www.mangod.top/articles/2024/04/24/1713970503394.html")

=====>>>>> [关于我](http://cxyroad.com/

”https://mp.weixin.qq.com/s/xHu3SS2fKqw7dvzNIGBLOQ”)
<<<<<=====

原文链接: <https://juejin.cn/post/7361284455536164899>