

使用Python语言实现 Appium自动化测试

APPIUM WITH PYTHON

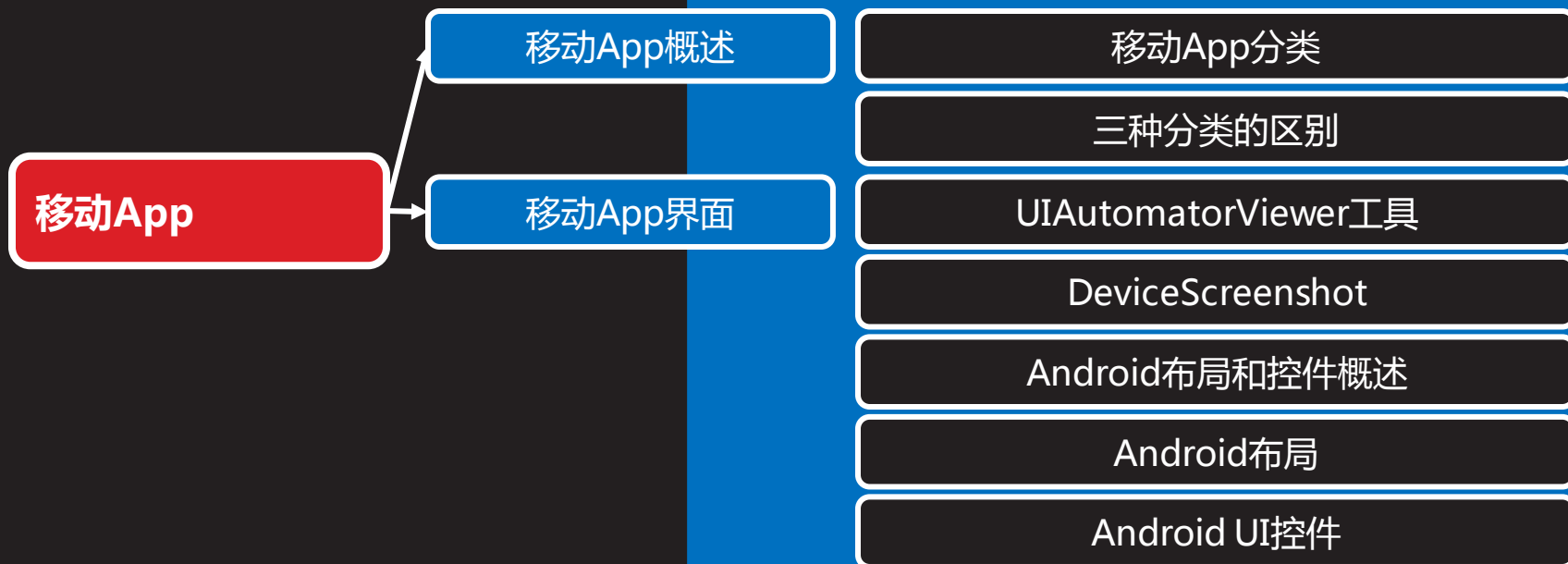
DAY04

内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	移动App
	10:30 ~ 11:20	
	11:30 ~ 12:20	Inspector
下午	14:00 ~ 14:50	
	15:00 ~ 15:50	
	16:00 ~ 16:50	
	17:00 ~ 17:30	总结和答疑



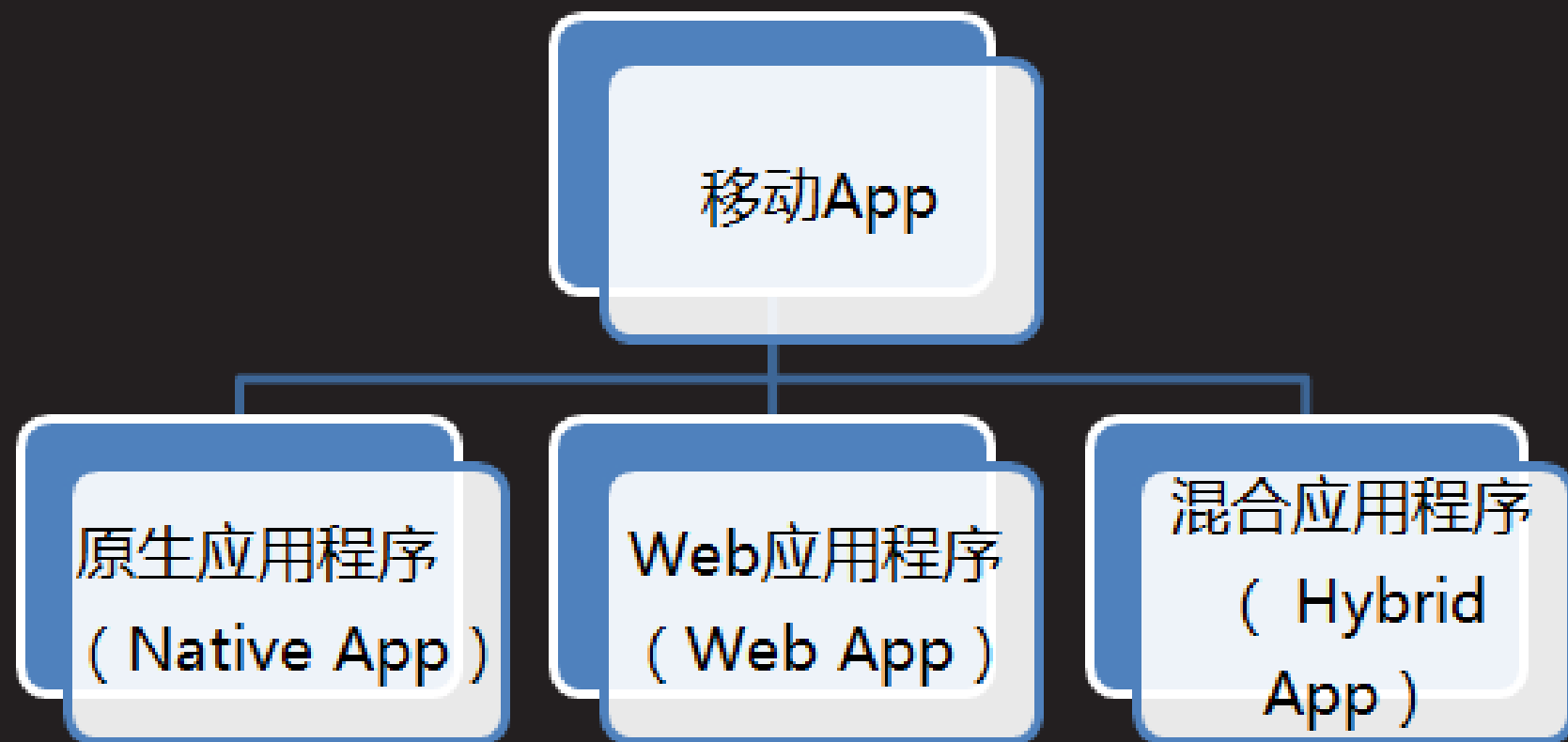
移动App



移动App概述



移动App分类



移动App分类（续1）

- 原生应用程序：
 - 原生应用程序是某一个移动平台（比如iOS或安卓）所特有的
 - 使用相应平台支持的开发工具和语言（比如iOS平台支持Xcode和Objective-C，安卓平台支持Eclipse和Java）。
 - 原生应用程序看起来（外观）和运行起来（性能）是最佳的。
- Web应用程序（HTML5应用程序）：
 - HTML5应用程序使用标准的Web技术，通常是HTML5、JavaScript和CSS。
 - 这种只编写一次、可到处运行的移动开发方法构建的跨平台移动应用程序可以在多个设备上运行。
- 混合应用程序：
 - 把HTML5应用程序嵌入到一个细薄的原生容器里面。

三种分类的区别

	Native App	Web App	Hybrid App
开发成本	高	低	中
维护更新	复杂	简单	简单
体验	优	差	优
StoreOrMarket认可	认可	不认可	认可
安装	需要	不需要	需要
跨平台	差	优	优

知识讲解

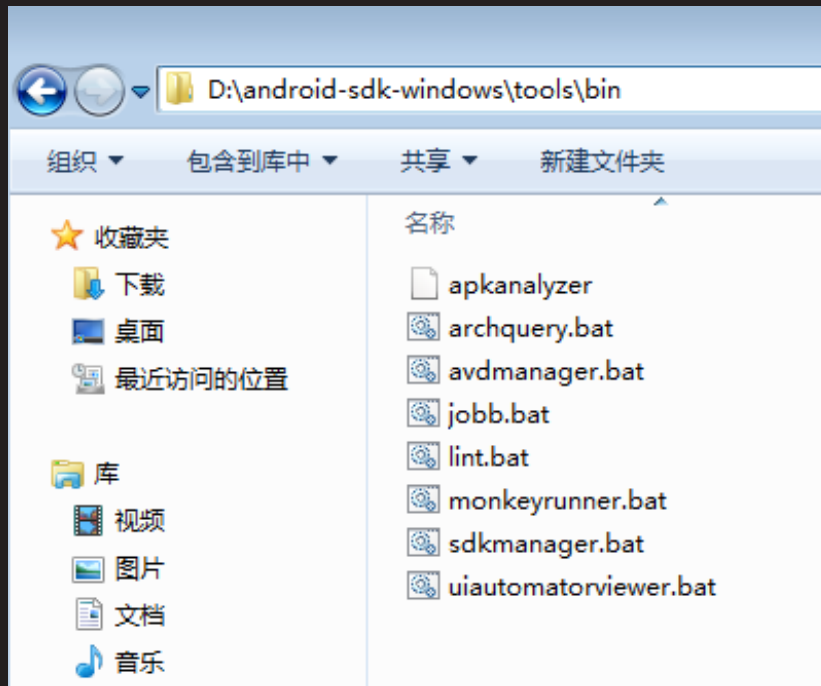


移动App界面



UIAutomatorViewer工具

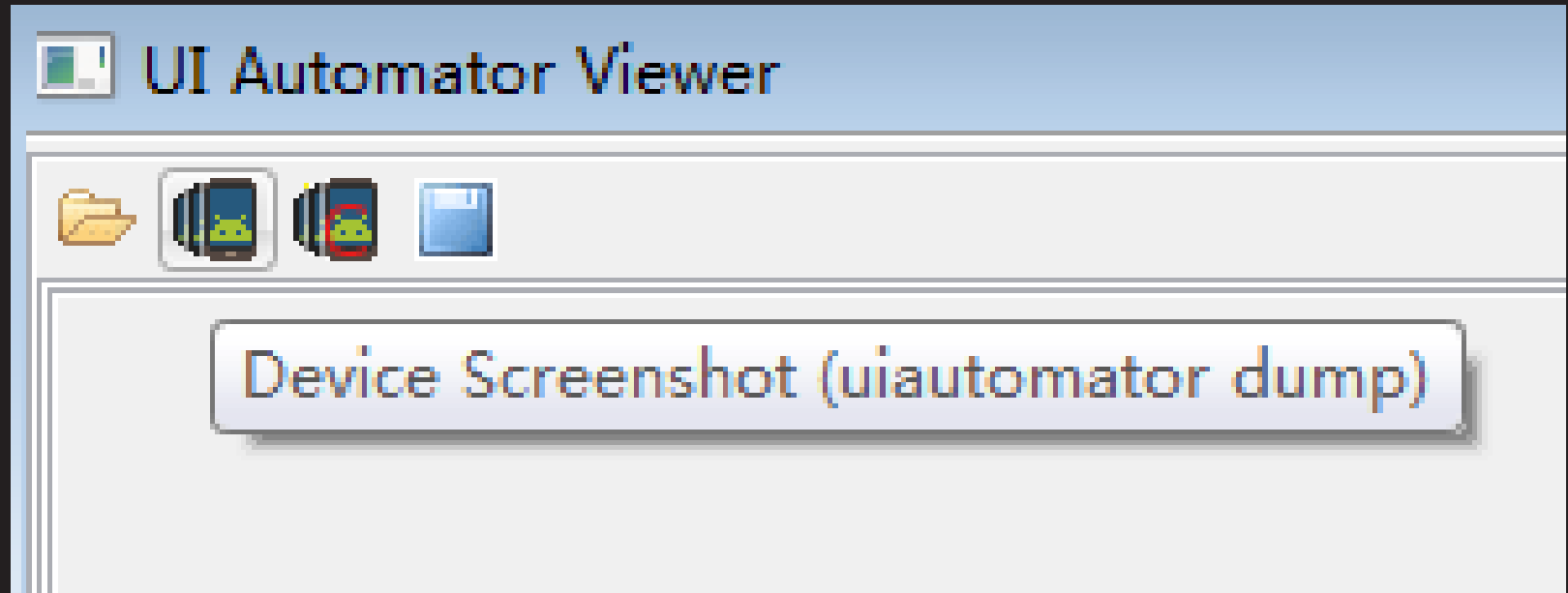
- 针对原生应用程序和混合应用，使用uiautomatorviewer查看界面上的元素。
- uiautomatorviewer是一个SDK自带的用来扫描和分析Android应用程序的UI组件的GUI工具。
- 位于SDK的Tools的bin目录下。



D:\android-sdk-windows\tools\bin\
uiautomatorviewer.bat

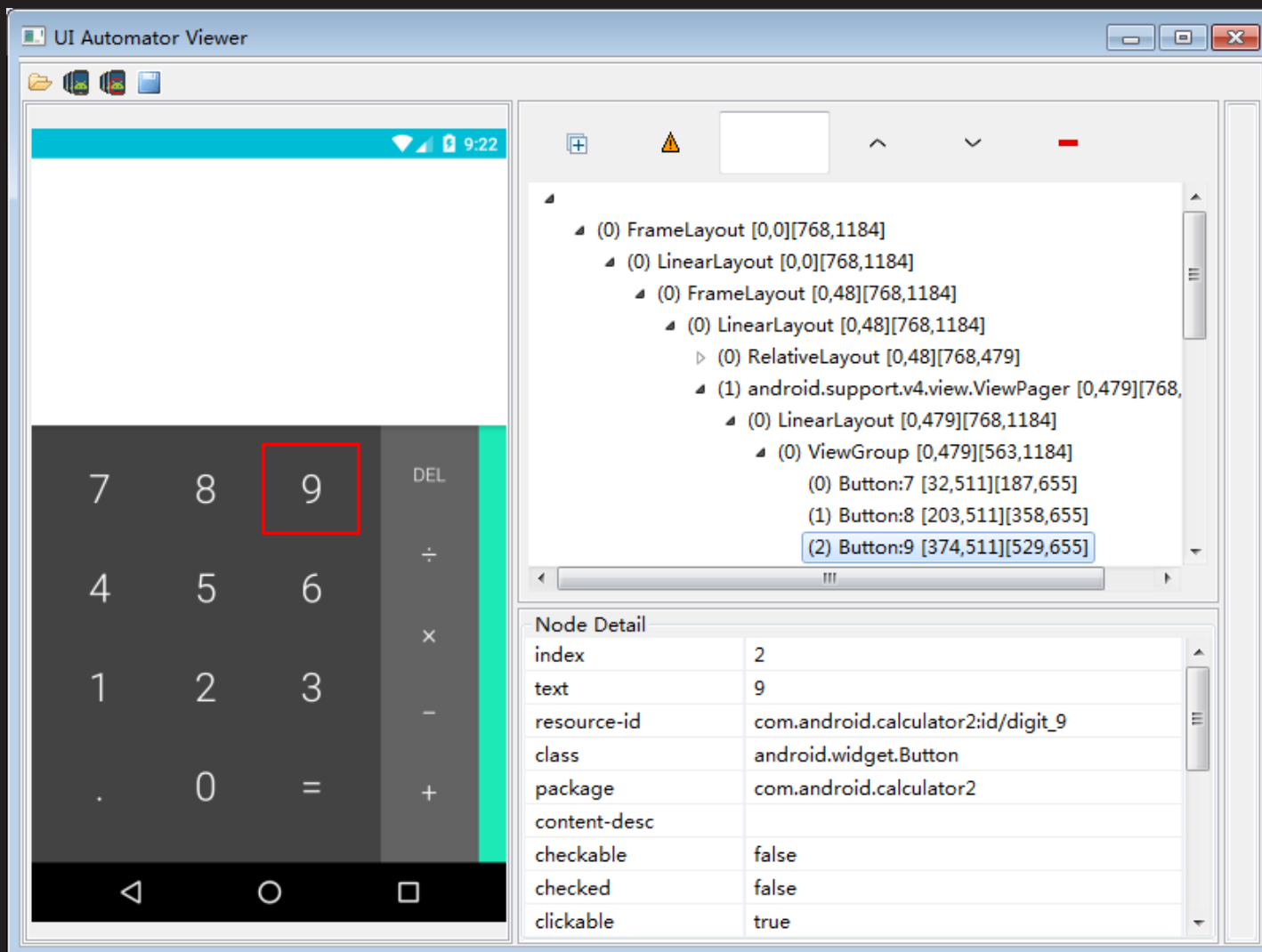
DeviceScreenshot

- UI Automator Viewer使用方法：
 - windows环境下，先打开1个模拟器，然后双击打开android-sdk目录下的tools下的uiautomatorviewer.bat文件
 - 在模拟器中手动打开某App
 - 点击UI Automator Viewer界面左上角的第二个按钮Device Screenshot (uiautomator dump)，就可以看到App界面



DeviceScreenshot (续1)

- 点击指定的控件，显示该页面元素的层级和各个属性值。



Android布局和控制件概述

- 布局：
 - 让控件在界面中按照设计的方式排列，本身不会包含内容，只是对控件位置的控制。
 - 不可见的容器（ ViewGroup ）定义 UI 的可视化结构，通过布局参数（ LayoutParams ）定义子元素的尺寸、位置。
- 控件：
 - 显示内容的组件，比如按钮的文字，或者图片等等，它包含了具体的内容并可以显示出来。



布局分类

线性布局 (LinearLayout)

常用

相对布局 (RelativeLayout)

常用

帧布局 (FrameLayout)

表格布局 (TableLayout)

网格布局 (GridLayout)

绝对布局 (AbsoluteLayout)



Android布局（续1）

- 线性布局（LinearLayout）：控件在线性方向上一次排列
 - 有水平线性布局和垂直线性布局之分。
 - 分为vertical（垂直）和horizontal（水平）。
 - 一般系统默认horizontal。
 - 例如：`android.widget.HorizontalScrollView`
- 相对布局（RelativeLayout）：通过相对定位的方式让控件出现在布局的任何位置
 - Android创建工程的时候一般默认相对布局。
 - 相对布局有两种形式，一种相对于容器而言，一种是相对于控件而言的。



Android布局（续2）

- 帧布局（FrameLayout）：子类会出现在布局的左上角
 - 帧布局设计的界面，只能在屏幕左上角显示一个控件。
 - 如果添加多个控件，这些控件回按照顺序在屏幕的左上角重叠显示，且会透明显示之前控件的文本，常见的刮刮卡就是通过帧布局实现的。
 - 例如：`android.widget.FrameLayout`
- 表格布局（TableLayout）：像表格一样有标准的行和列
 - 行数由TableRow对象控制，即布局中有多少TableRow对象，就有多少行。
 - 列数由最宽的单元格决定，假如第一个TableRow有两个控件，第二个TableRow有三个控件，那么这个TableRow就有三个控件。



Android布局（续3）

- 网格布局（GridLayout）：类似于表格布局，但比表格布局更灵活，可以比较随意的调整其中的分区模块
 - 用一组无限细的直线将绘图区域分成行、列和单元，并指定控件的显示区域和空间在该区域的显示方式。
- 绝对布局（AbsoluteLayout）：使用绝对坐标来堆放组件
 - 通过指定x、y坐标来确定每一个空间的位置。
 - 控件以屏幕左上角为坐标原点。



Android UI控件

知识讲解

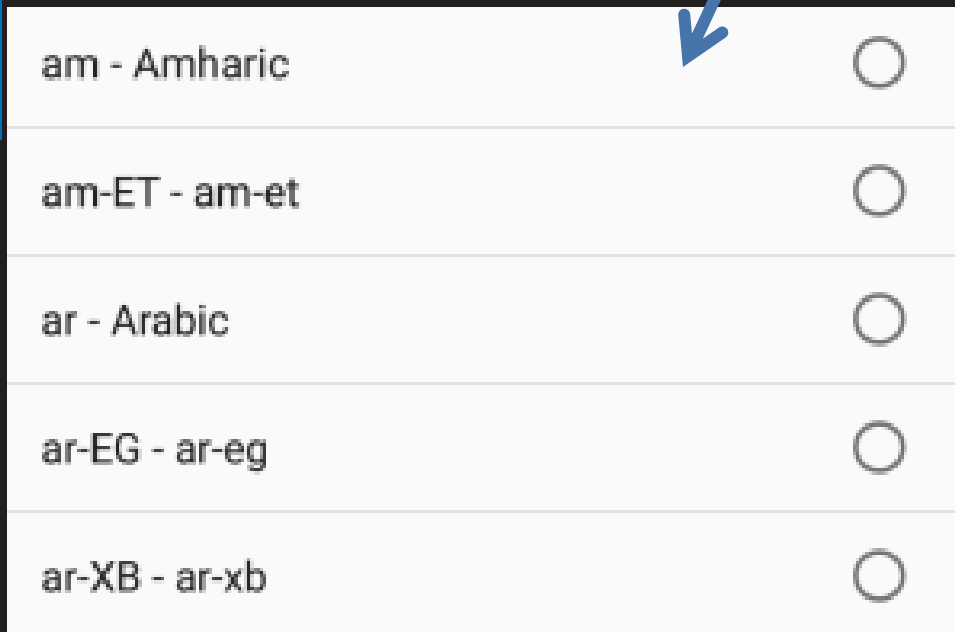
	控件	说明
1	android.widget.TextView	显示文字，用来展示文本
2	android.widget.EditText	输入框，可编辑，可设置软键盘方式。
3	android.widget.Button	按钮（上面显示文本）
4	android.widget.ImageButton	图片按钮
5	android.widget.ImageView	显示图片
6	android.widget.CheckBox	复选框
7	android.widget.RadioButton	单选按钮
8	android.widget.ToggleButton	开关形式的按钮
9	android.widget.DatePicker	日期选择控件
10	android.widget.TimePicker	时间选择控件
11	android.widget.Spinner	下拉框中选择单项值控件



Android UI控件（续1）

	控件	说明
12	android.widget.Switch	开关
13	android.widget.ListView	列表
14	android.widget.CheckedTextView	列表单选或多选控件

知识讲解



Inspector

Inspector概述

Inspector使用

Appium Desktop工具

启动Inspector

Desired Capabilites参数

保存Capability集合

启动会话

Inspector窗口

工具栏

选择元素

查看定位信息

录制脚本

按坐标滑动

按坐标触摸

后退

刷新

搜索元素

复制源码到剪切板

关闭会话

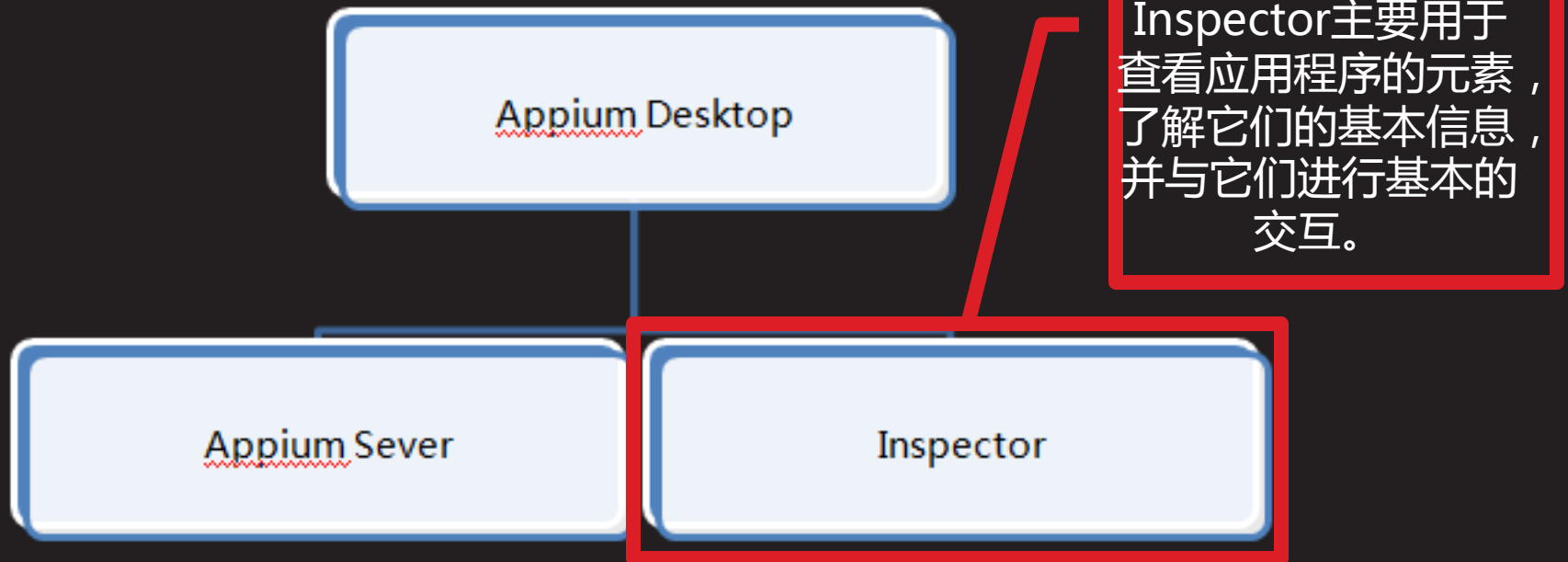
Inspector

Inspector概述



Appium Desktop工具

- Appium Desktop (Appium桌面) 是一款适用于Mac、Windows和Linux的应用程序，它为您提供了Appium自动化服务器在漂亮灵活的用户界面中的强大功能。
- 分为：Appium Server 和 Inspector。

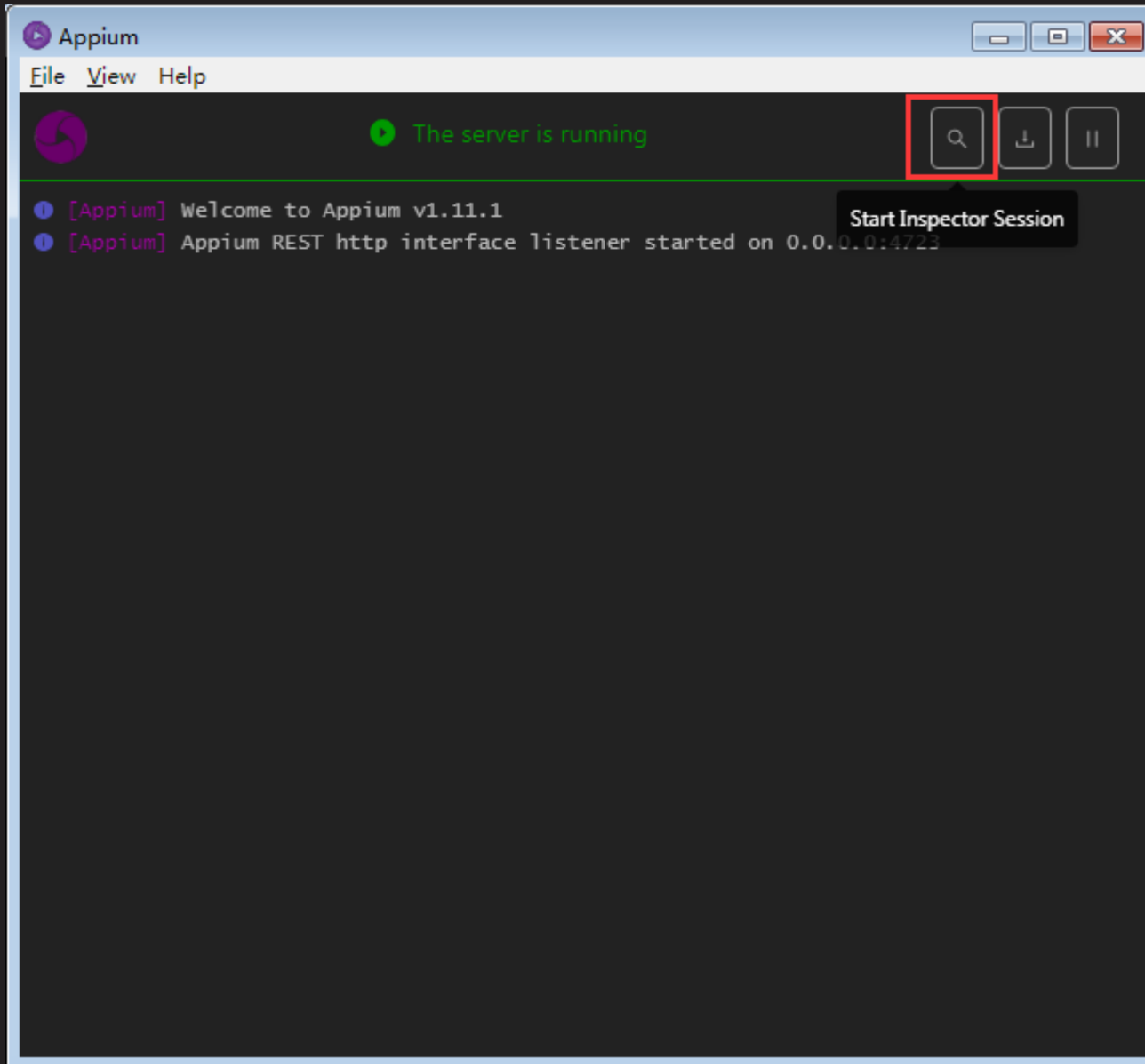


启动Inspector

- 启动Inspector :
 - 启动Appium Desktop , 选择Simple模式即可
 - 点击Start Server启动Appium服务
 - Start Inspector Session(点击 放大镜按钮)



启动Inspector (续1)



Desired Capabilities参数

- 在这个窗口我们需要配置一些Desired Capabilities信息。

Field	Type	Value	Action
platformName	text	Android	Trash
platformVersion	text	4.3	Trash
deviceName	text	android Simulator	Trash
app	filepath	D:\app\SHIPPER_A	Trash
appPackage	text	com.taobao.taobao	Trash
appActivity	text	com.taobao.tao.welcom	Trash

JSON Representation

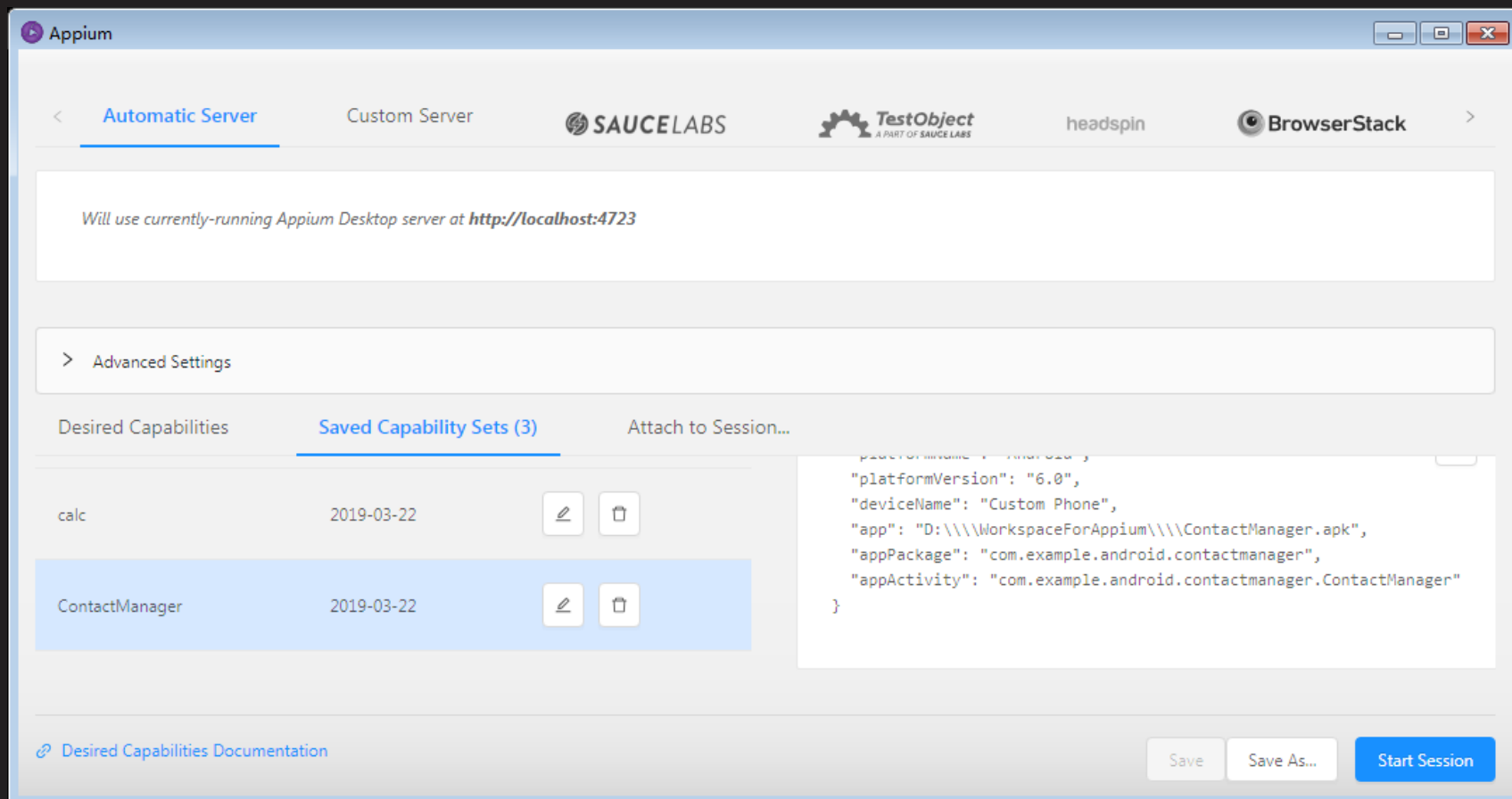
```
{
  "platformName": "Android",
  "platformVersion": "4.3",
  "deviceName": "android Simulator",
  "app": "D:\\app\\SHIPPER_Android_TEST_4.1.1.0.apk",
  "appPackage": "com.taobao.taobao",
  "appActivity": "com.taobao.tao.welcome.Welcome"
}
```

注意：

- 1、app选择filepath类型。
- 2、如果想要使用真机，必须再加一个字段udid，是text类型。
- 3、unicodeKeyboard和resetKeyboard，是bool类型。

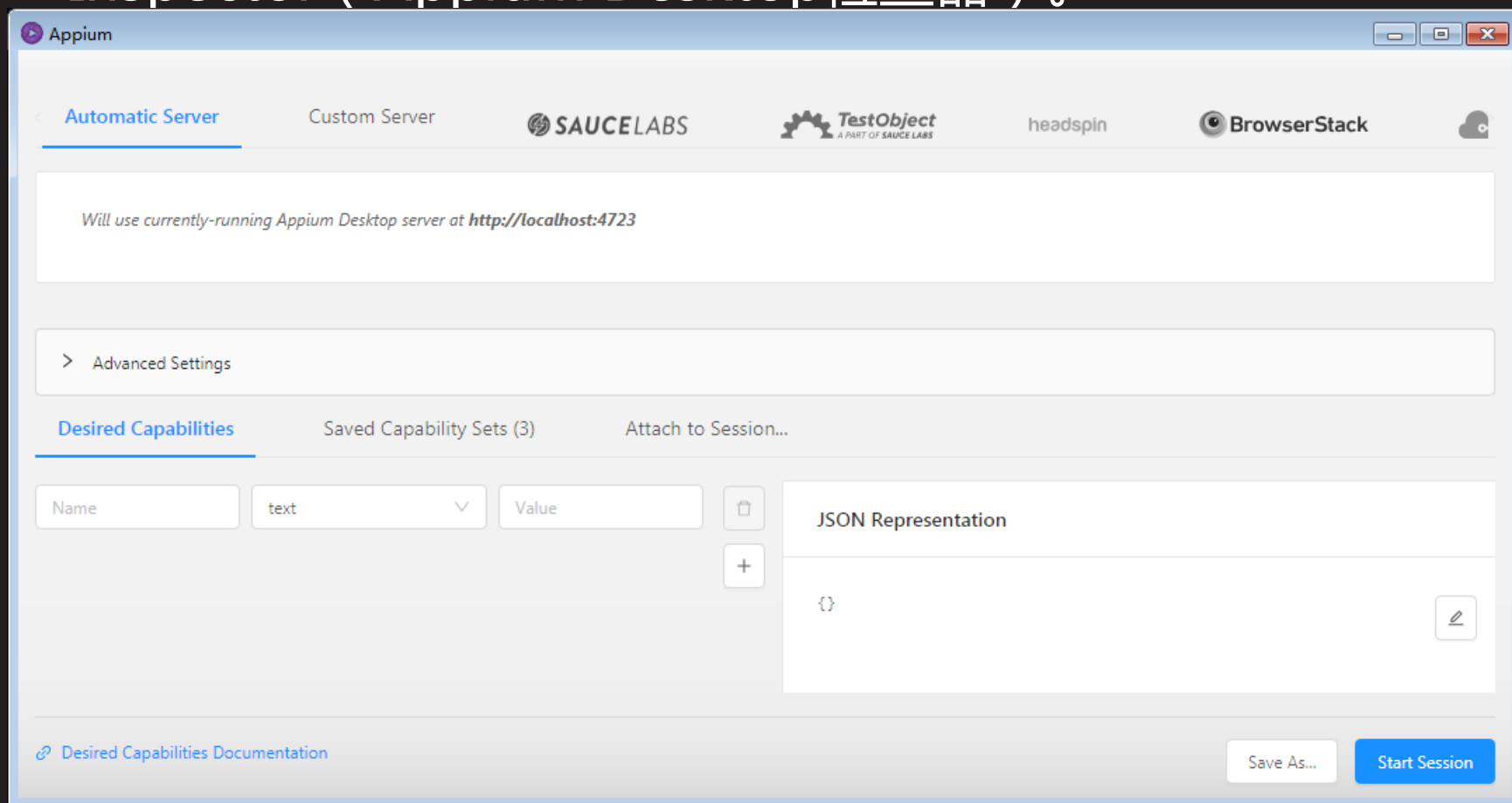
保存Capability集合

- 输入Desired Capabilites各个参数后，点击Save As，输入名称保存后，可以在Saved Capability Sets中查看并快速启动会话。



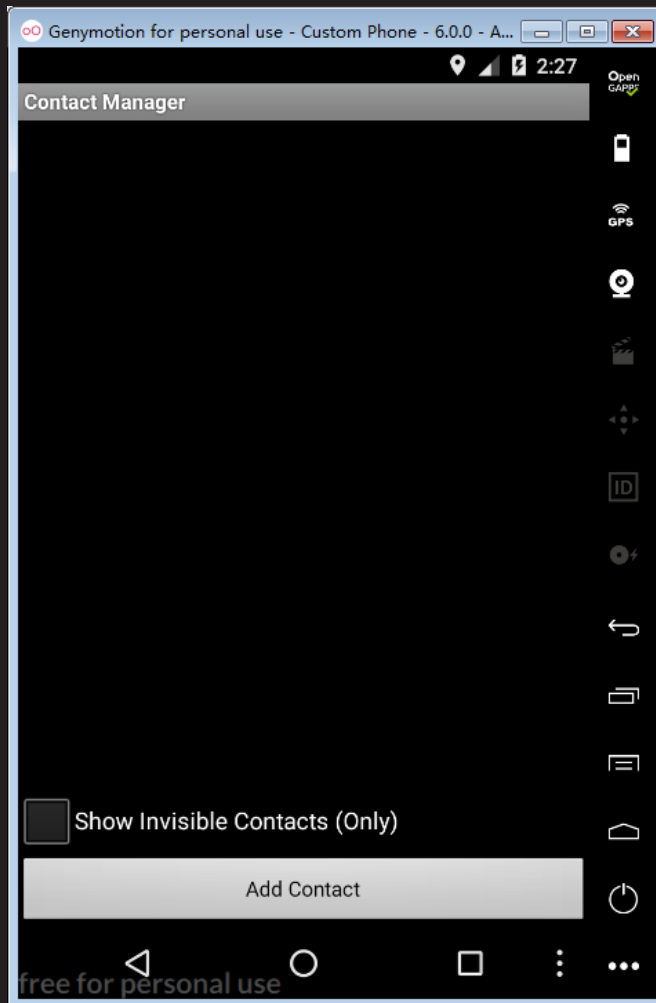
启动会话

- 设置好服务器type和capabilities后，单击“Start Session”（启动会话）来启动Appium Desktop Inspector（Appium Desktop检查器）。



启动会话（续1）

- 启动会话后，手机或模拟器会自动启动Desired Capabilities参数所指定的App。



启动地图App会话

- 启动地图App会话，详见【COOKBOOK】

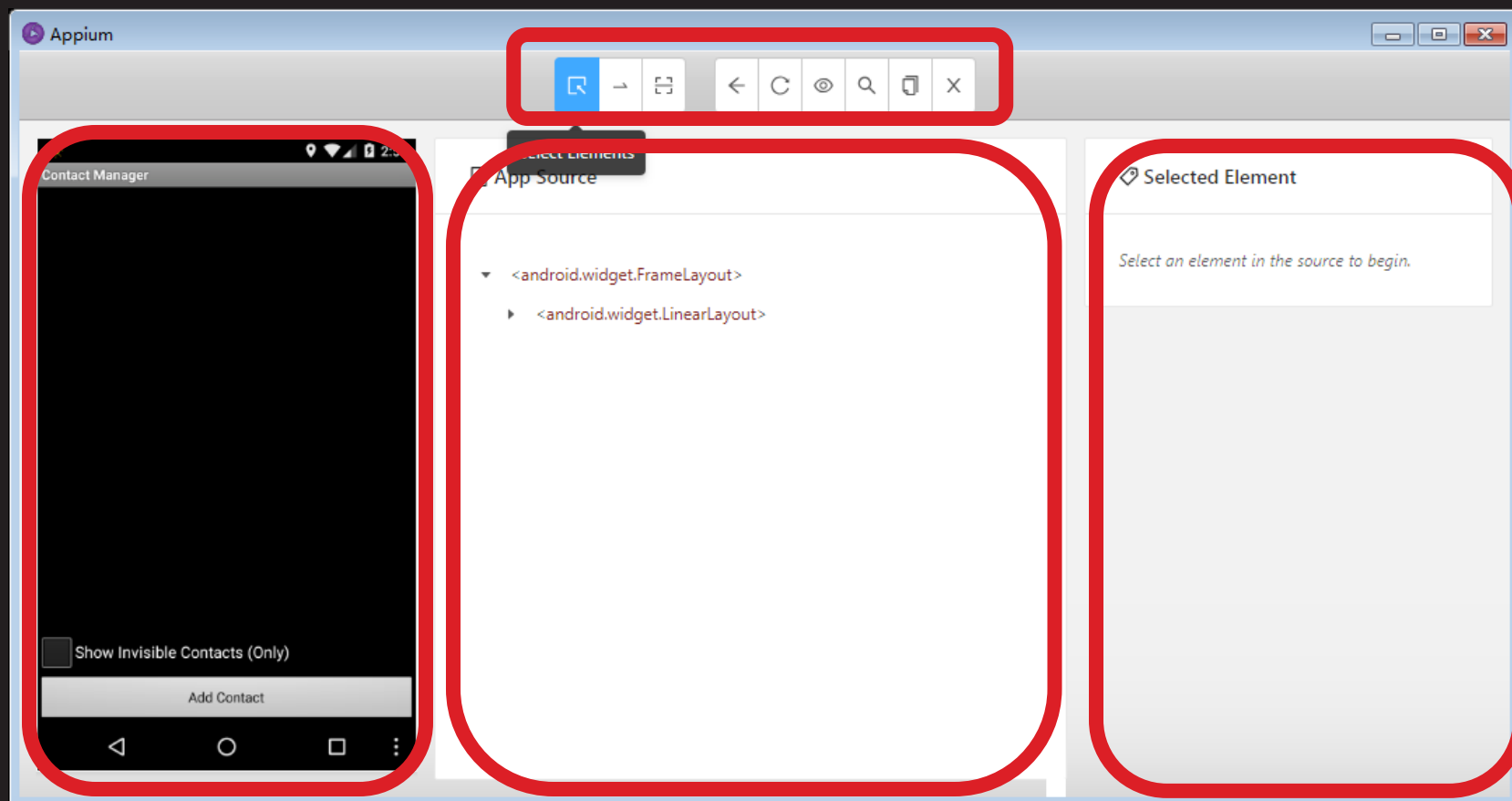


Inspector工具使用



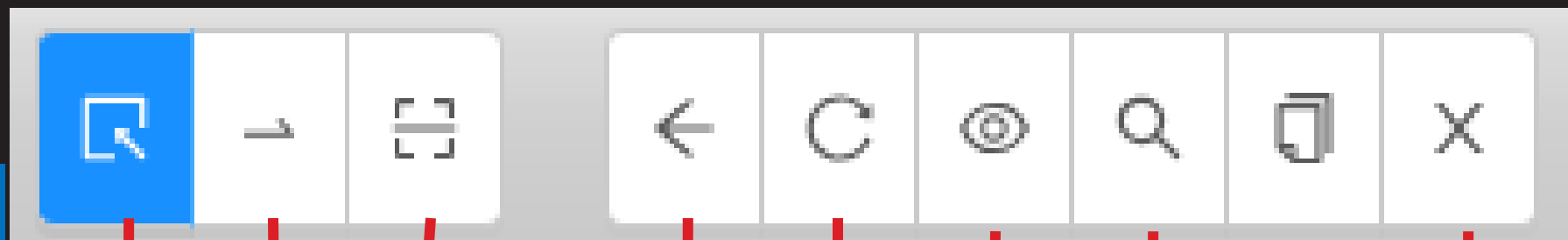
Inspector窗口

- Appium Inspector元素分析的新窗口上方有一排工具栏，下方主页面分为左中右三部分：App界面、App源码、元素信息。



工具栏

- 工具栏说明：



知识讲解

Select
Elements
选择元素

Tap By
Coordinates
按坐标触
摸

Refresh
Source
& Screens
刷新源
码和截屏

Tap By
Coordinates
按坐标触
摸

Quit
Session
& Close
Inspector
关闭

Swipe By
Coordinates
按坐标滑
动

Back
返回

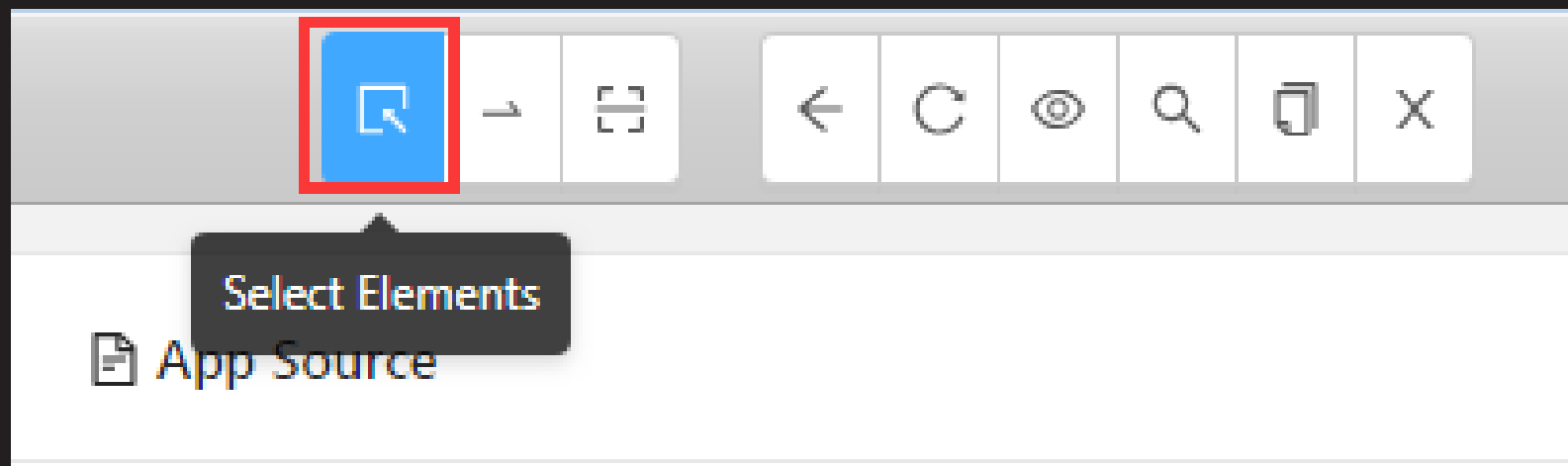
Start
recording
开始录制

Search For
Element
搜索元素



选择元素

- 选择元素的步骤：
 - 点击界面上方选择元素（ Select Elements ）的按钮。
 - 再点击左侧App中的某个元素。
 - 当元素突出显示时，其对应的一行页面源码信息将高亮显示在中部的App源码视图中，并且其属性和定位信息显示在右侧视图中。



选择元素 (续1)

The screenshot displays the Appium web interface. On the left, a mobile app preview titled 'Contact Manager' shows a button labeled 'Show Invisible Contacts (Only)'. A red rectangle highlights this button, with a red arrow pointing to the 'App Source' panel. The 'App Source' panel shows the XML hierarchy of the app's UI. A red rectangle highlights the following XML snippet:

```
<android.widget.CheckBox content-desc="Show Invisible Contacts (Only)" />
```

Another red arrow points from this snippet to the 'Selected Element' panel on the right. This panel shows the 'Find By' and 'Selector' information for the selected element.

Find By	Selector
accessibility id	Show Invisible Contacts (Only)
id	com.example.and roid.contactmana ger:id/showInvisi ble
xpath	//android.widget. CheckBox[@cont ent-desc="Show Invisible Contacts (Only)"]

Attribute	Value
index	1



查看定位信息

- 查看元素定位信息，就可以书写定位元素的代码。
- 例如：

知识讲解

Find By	Selector
accessibility id	Show Invisible Contacts (Only)
id	com.example.android.contactmanager:id/showInvisible
xpath	//android.widget.CheckBox[@content-desc="Show Invisible Contacts (Only)"]

`driver.find_element_by_accessibility_id("Show Invisible Contacts (Only)").click()`

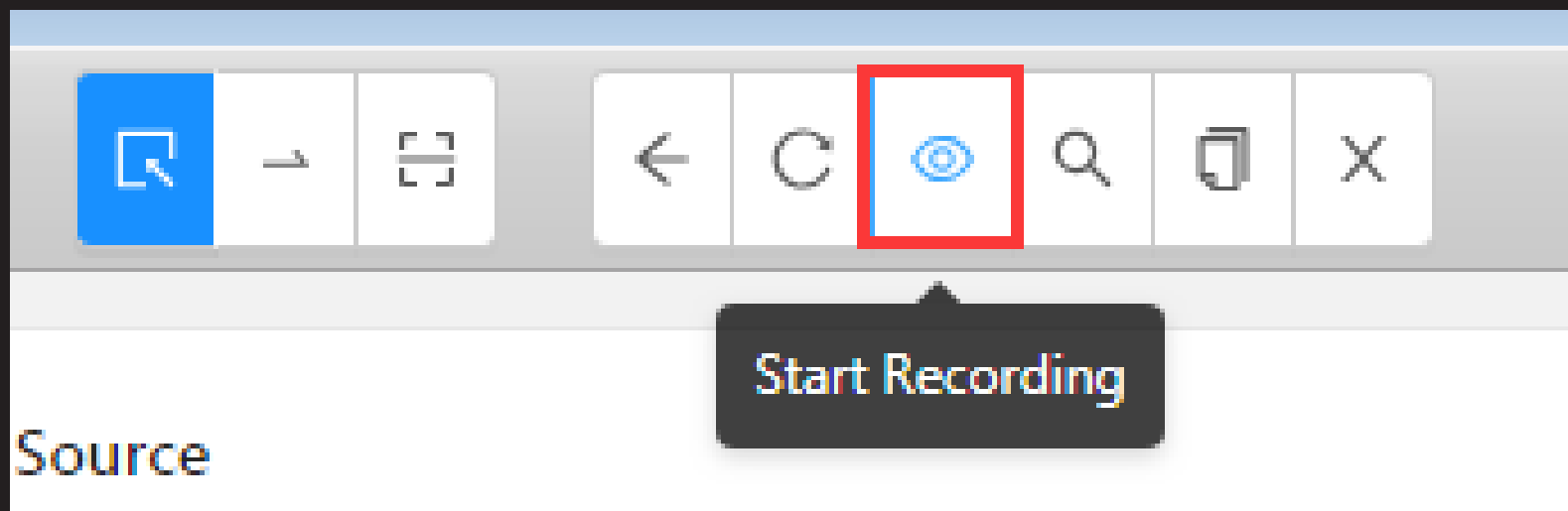
`driver.find_element_by_id("com.example.android.contactmanager:id/showInvisible").click()`

`driver.find_element_by_xpath("//android.widget.CheckBox[@content-desc=\"Show Invisible Contacts (Only)\"]").click()`



录制脚本

- 录制脚本是一个很实用的功能，解决了该使用哪个API的问题以及脚本的编写规范的问题。
- 对于初学者来说是一个很好的参考。
- 脚本录制功能就是那个眼睛图标，如下图：



录制脚本（续1）

- 录制脚本步骤：
 - 点击眼睛图标开始录制（Start Recording）。
 - 工具栏切换到选择元素（Select Elements）
 - 在左侧App中选择一个元素。
 - 在右侧元素定位视图选择相应的操作动作。
 - 选择脚本目标语言后，可以复制到剪切板。



录制脚本（续2）

- 录制脚本动作说明：
 - Tap触摸（点击）、Send Keys输入、Clear清空

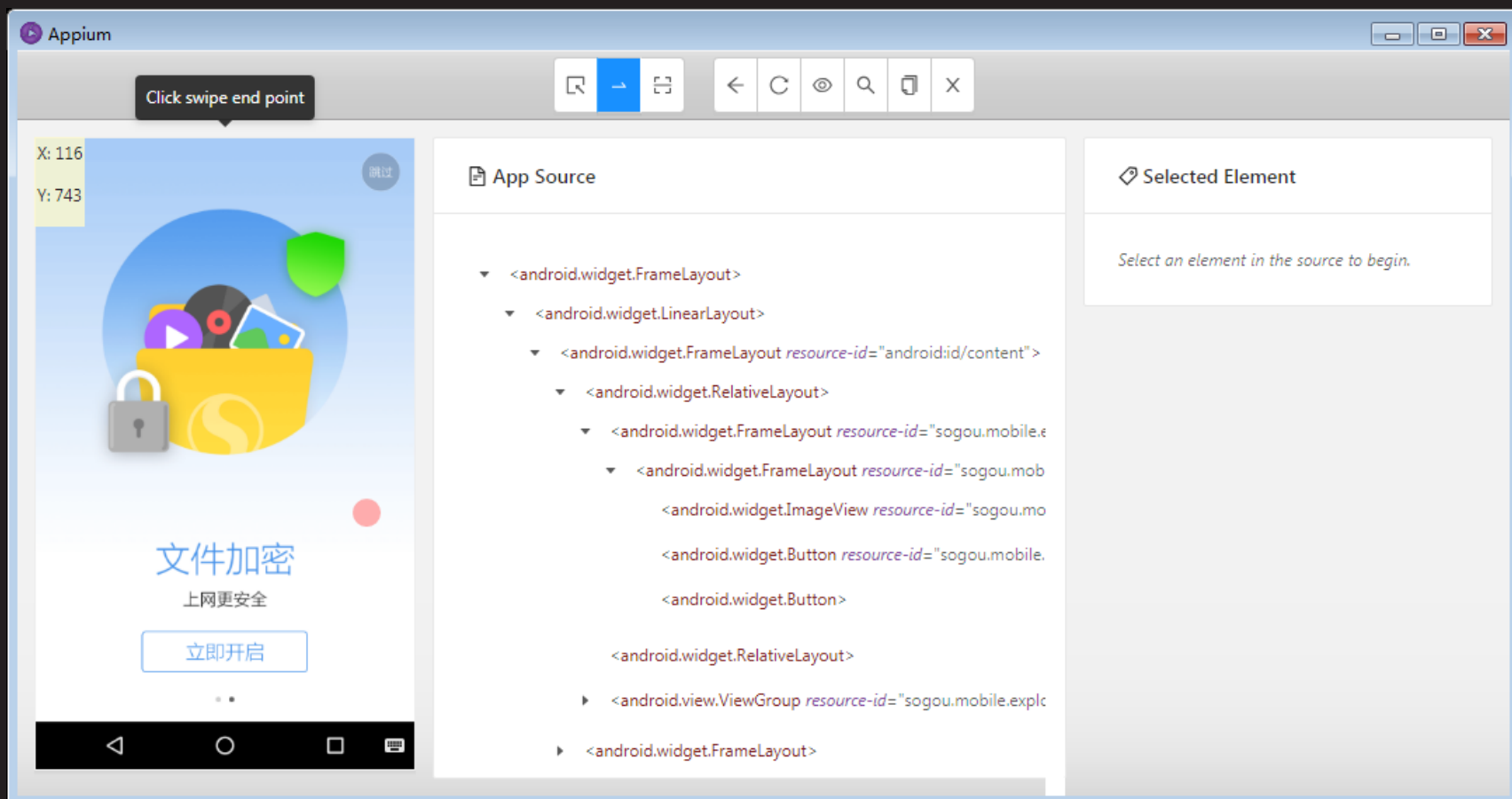
The screenshot displays the Appium web interface. On the left, a mobile app window titled 'Contact Manager' is shown with a button labeled 'Show Invisible Contacts (Only)'. A red box highlights this button, and a red arrow points from it to the 'Tap' button in the 'Selected Element' panel on the right. The 'Recorder' panel in the center shows a Python script: `e11 = driver.find_element_by_accessibility_id("Show Invisible Contacts (Only)")` and `e11.click()`. The 'App Source' panel shows the XML structure of the app. The 'Selected Element' panel on the right has a 'Tap' button highlighted with a red box, and below it, a table showing the element's attributes and selectors.

Find By	Selector
accessibility id	Show Invisible Contacts (Only)
id	com.example.and roid.contactmana ger:id/showInvisi ble
xpath	//android.widget. CheckBox[@cont ent-desc="Show Invisible Contacts (Only)"]

Attribute	Value
-----------	-------

按坐标滑动

- 选择工具栏第二个按钮（Swipe）后，点击界面上滑动操作的起始位置和结束位置，完成滑动操作。

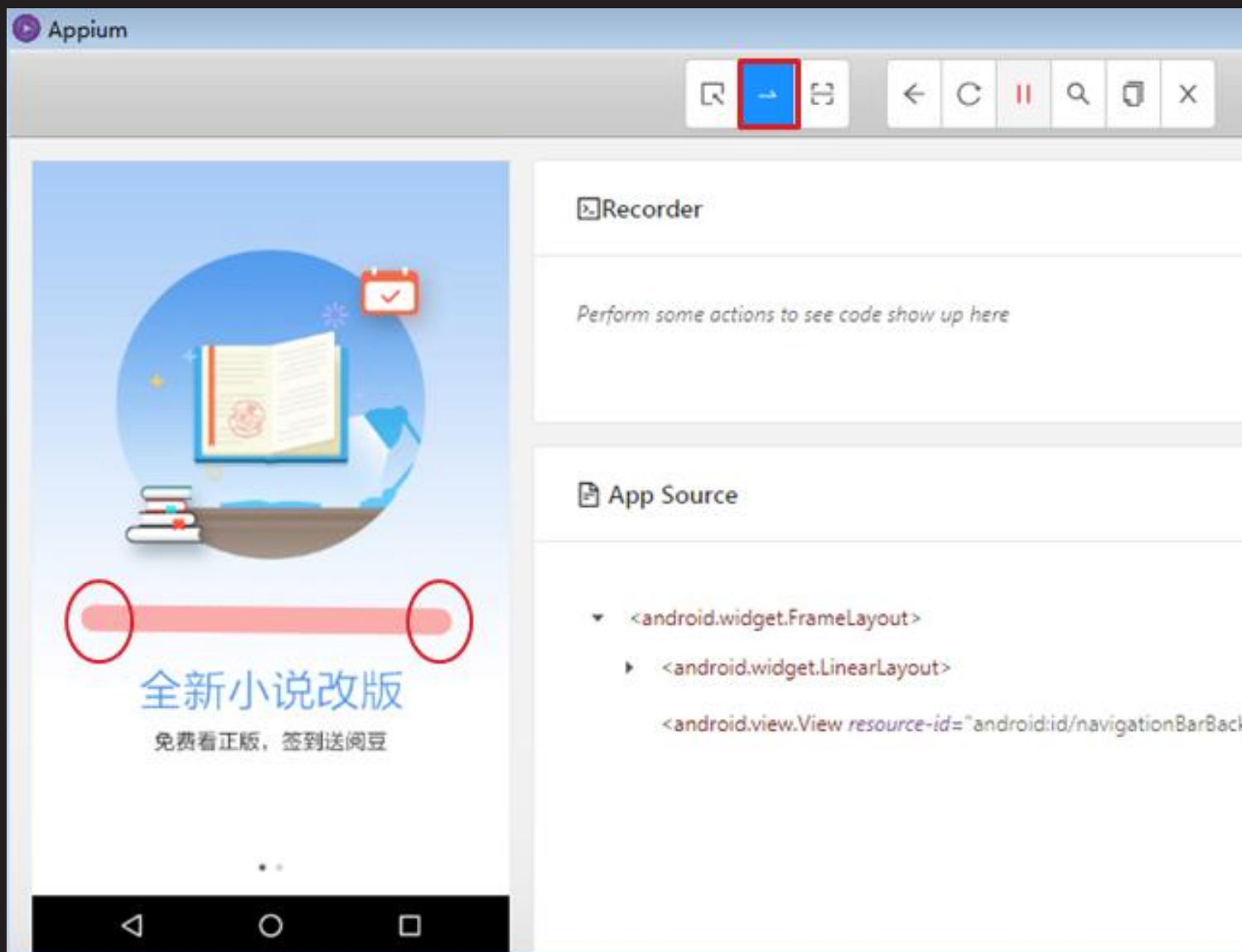


按坐标滑动（续1）

- 录制按坐标滑动的步骤：
 - 工具栏点击“开始录制”（Start Recording）
 - 工具栏切换到“按坐标滑动”（Swipe By Coordinates）
 - 在App界面中点击开始滑动的坐标位置（会标记为粉色圆点）
 - 在App界面中点击结束滑动的坐标位置（会标记为粉色直线）
 - 选择目标语言
 - 复制生成的代码
- 注意：执行代码前导入TouchAction
 - `from appium.webdriver.common.touch_action`
`import TouchAction`

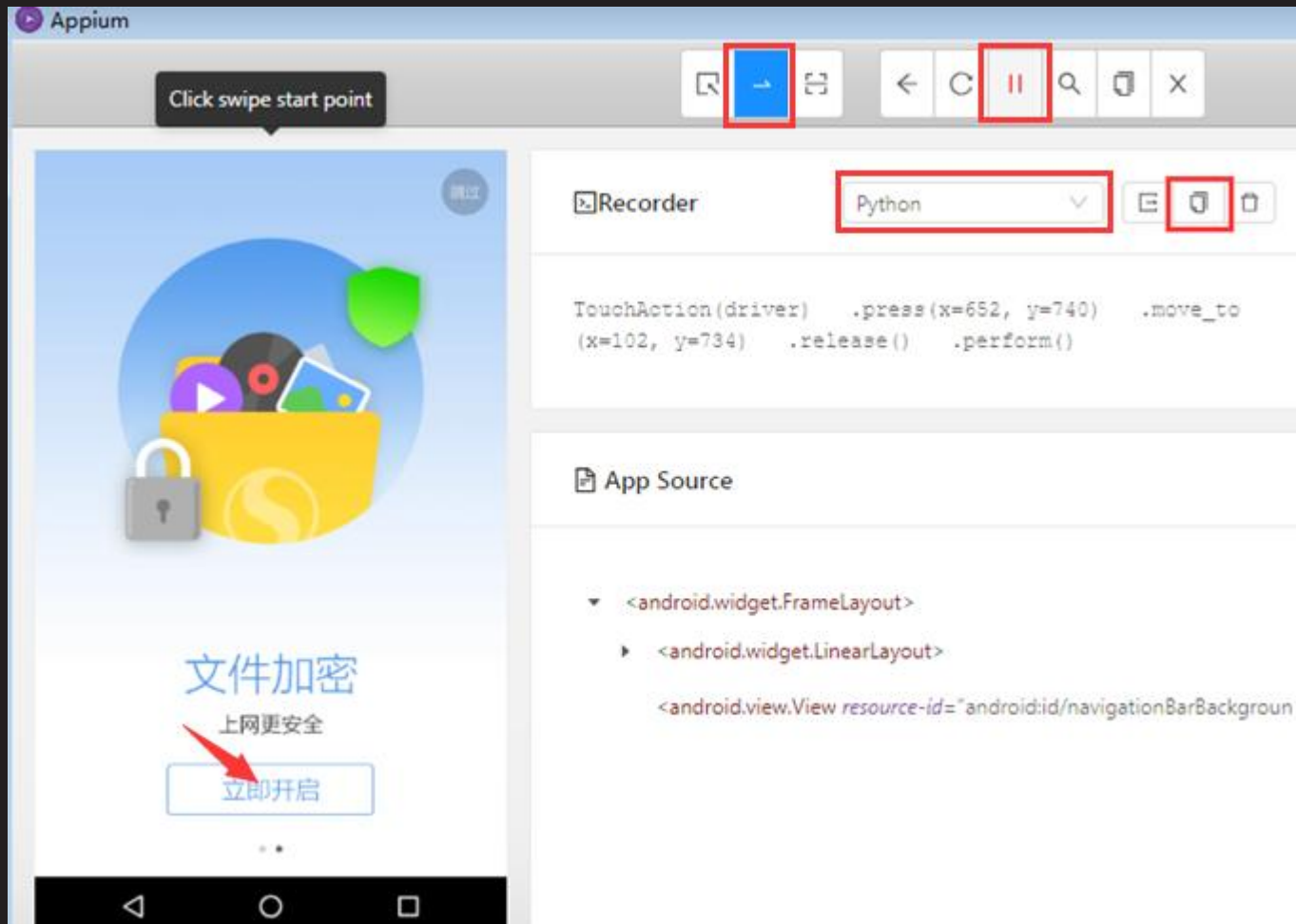


按坐标滑动（续2）



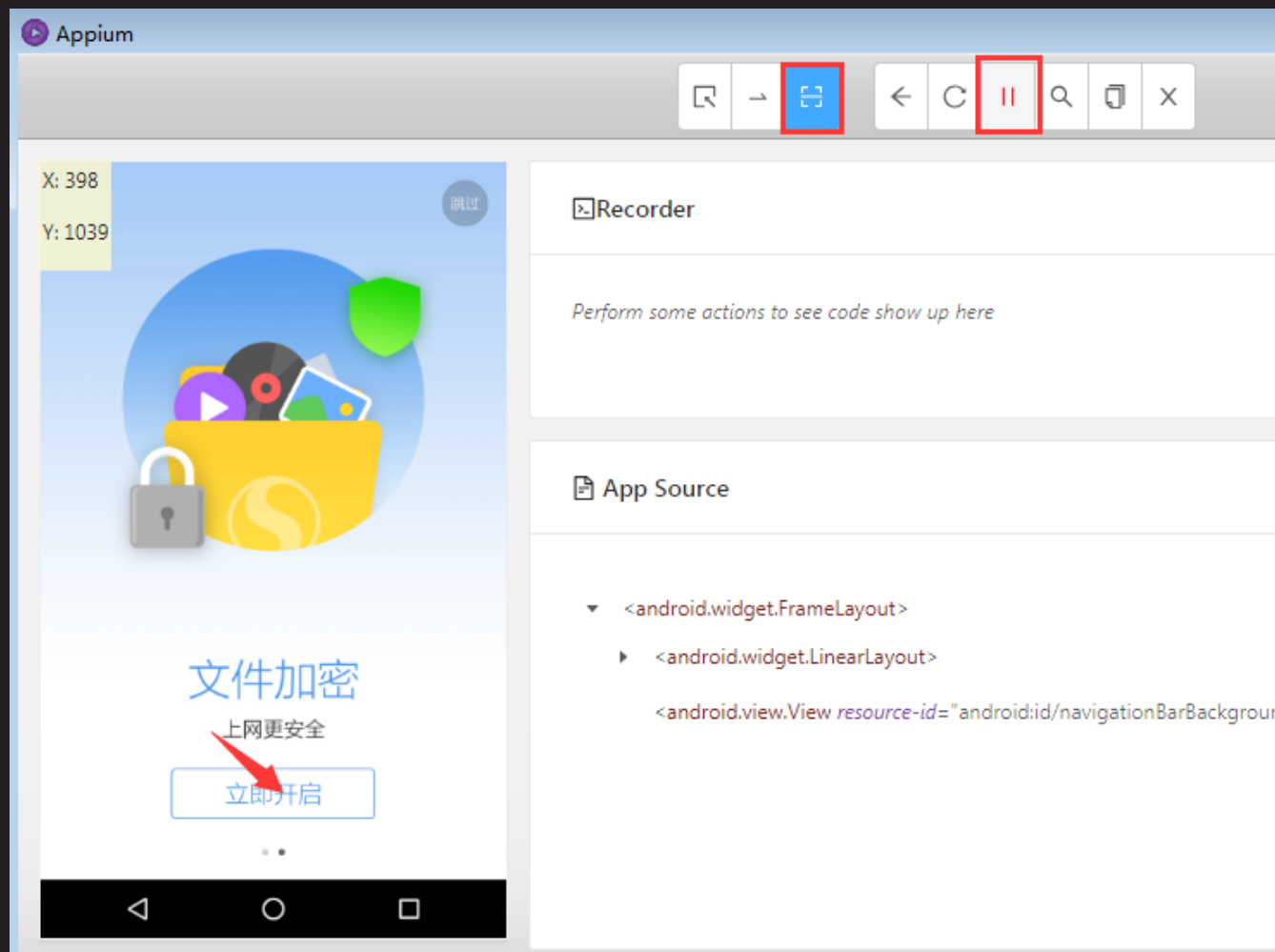
按坐标滑动（续3）

- 示例：TouchAction(driver).press(x=652, y=740).move_to(x=102, y=734).release().perform()



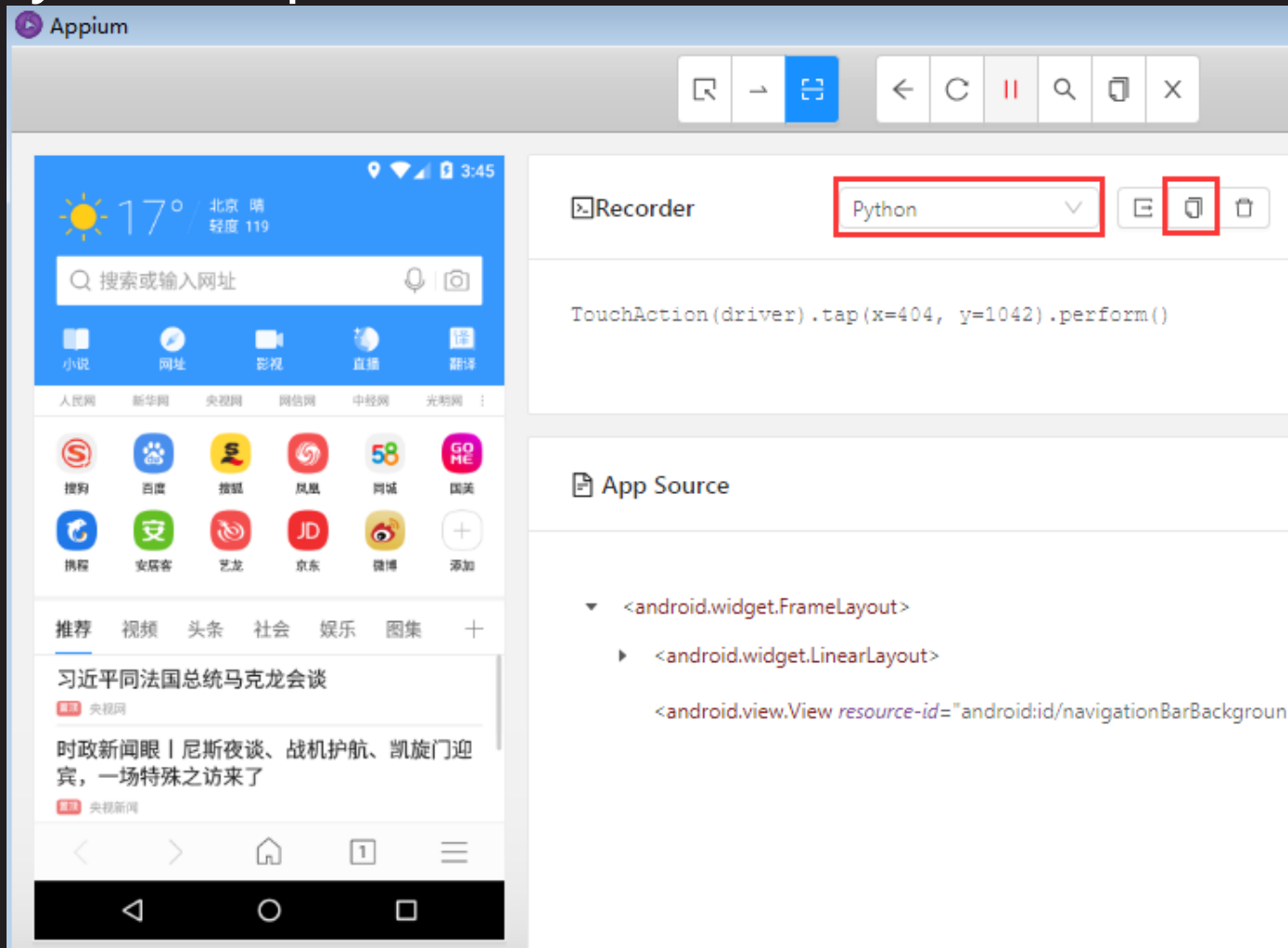
按坐标触摸

- 选择工具栏第三个按钮（Tap By Coordinates按坐标触摸）后，点击App界面上的触摸位置。



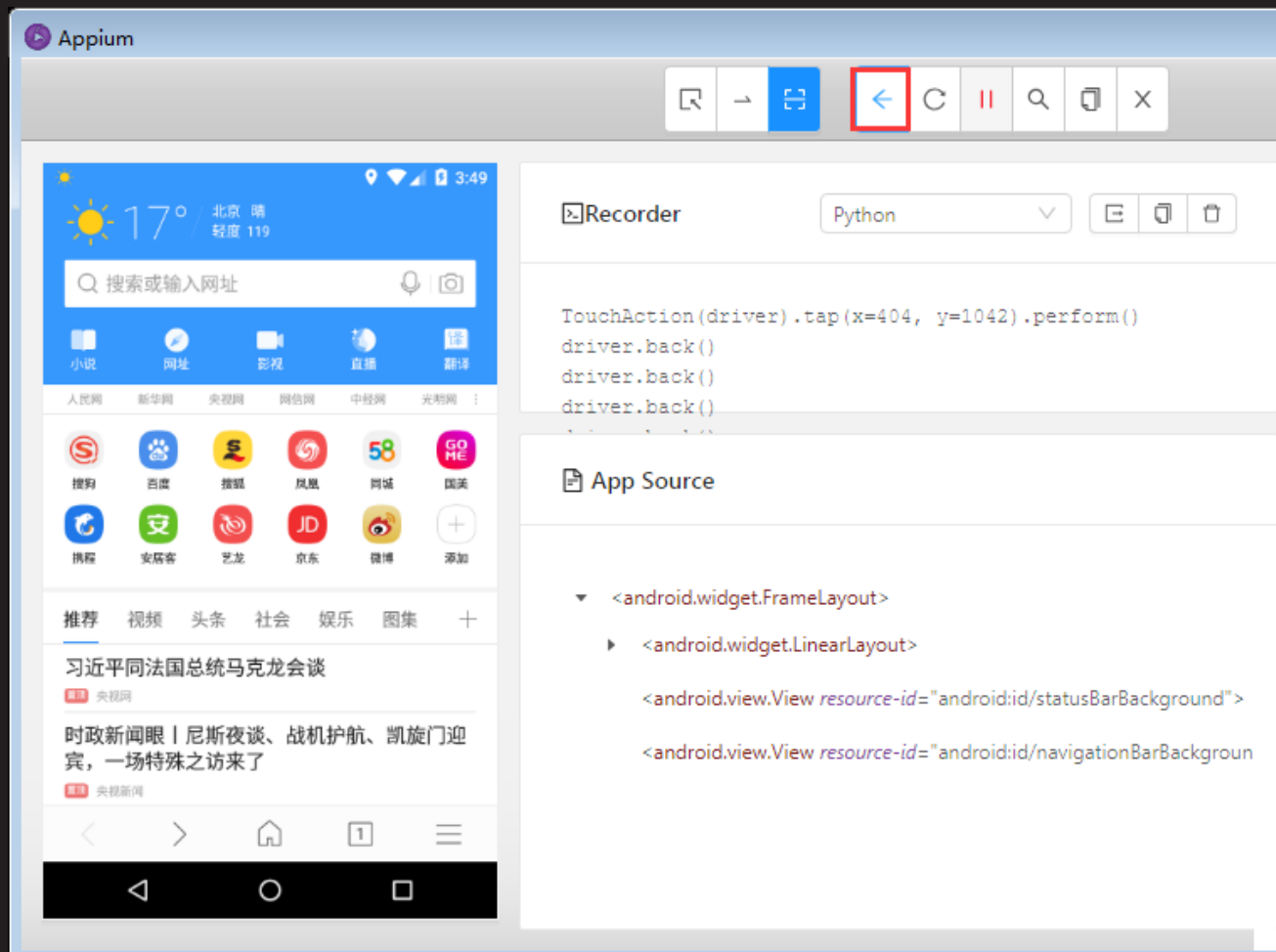
按坐标触摸（续1）

- 示例：TouchAction(driver).tap(x=404, y=1042).perform()

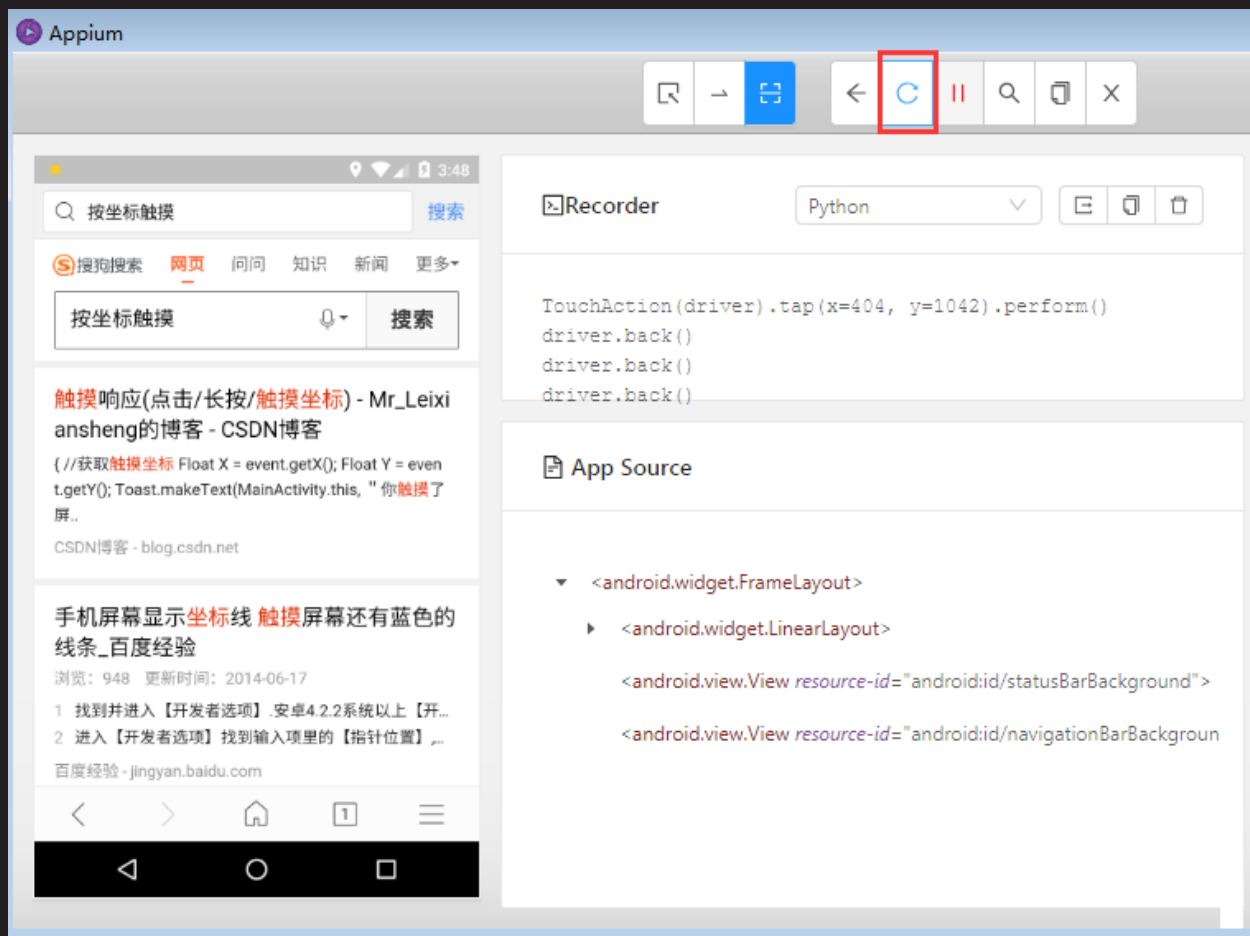


后退

- 后退 (Back) : 后退回前一个页面。

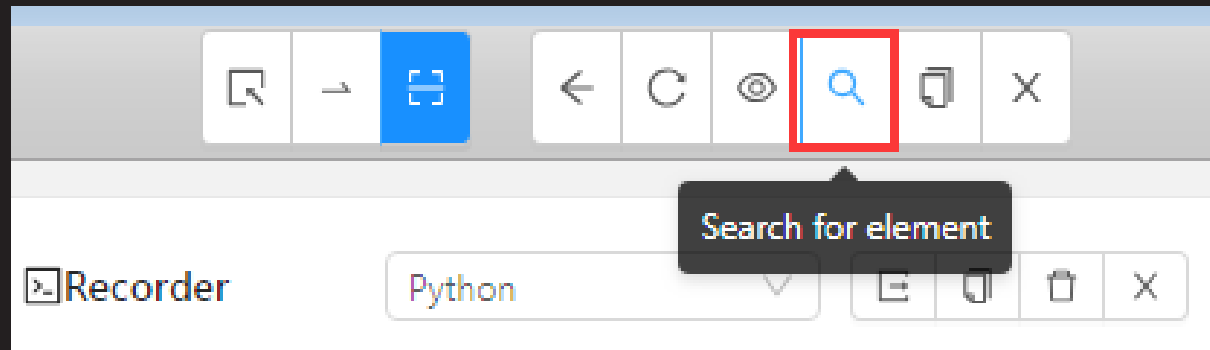


- 刷新 (Refresh Source&Screenshot) : 刷新当前App源码和截图



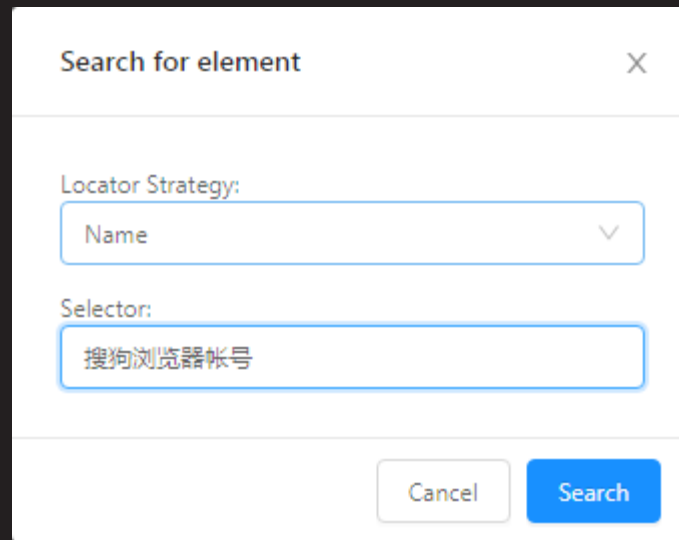
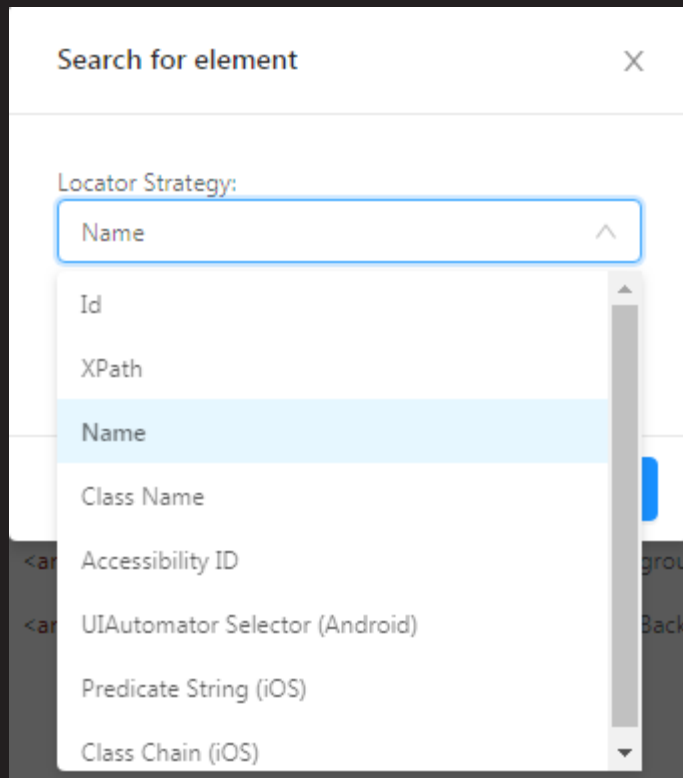
搜索元素

- 搜索元素（Search for element）：使用指定条件搜索元素。
 - 工具栏点击Search for element
 - 选择定位策略（Locator Strategy）
 - 输入搜索条件的数据（Selector）
 - 点击“搜索”按钮（Search）
 - 查看搜索结果并复制代码到剪切板



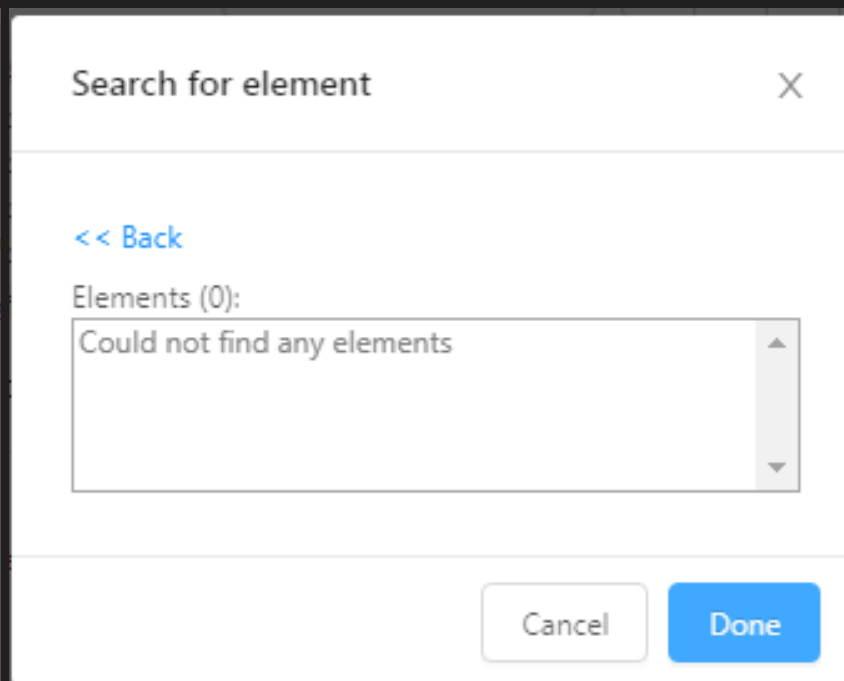
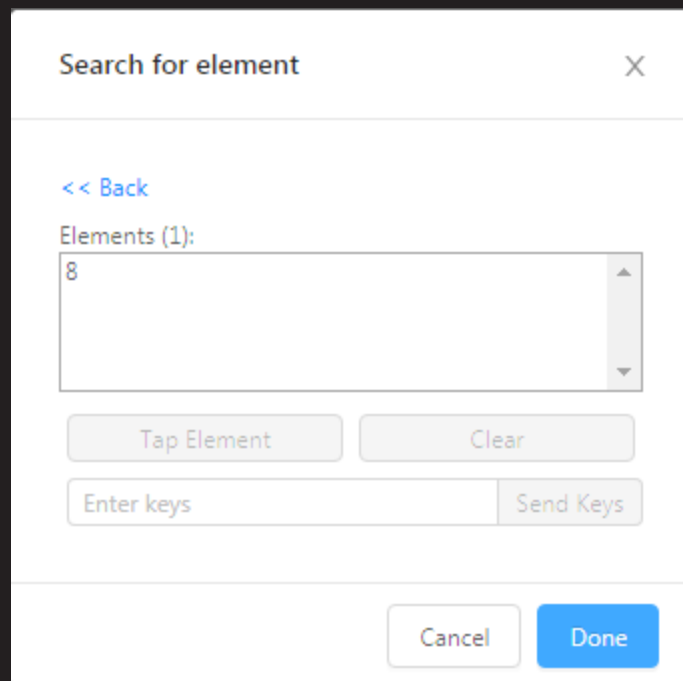
搜索元素（续1）

- 定位策略（Locator Strategy）有：Id、Xpath、Name、Class Name、Accessibility ID、UIAutomator Selector（Android）、Predicate String（iOS）、Class Chain（iOS）



搜索元素（续2）

- 搜索结果数显示在Elements后的小括号中。
 - 1代表有1个，0代表没有搜索到任何结果。
 - 如果结果数不等于0，最后生成的测试代码可以获得由搜索结果的WebElement组成的列表（List）。



搜索元素（续3）

The screenshot displays the Appium web interface. On the left, a mobile app interface is shown with a login screen titled '登录帐号' (Login Account). It features four login options: '搜狗浏览器帐号' (Sogou Browser Account), '微信帐号' (WeChat Account), 'QQ帐号' (QQ Account), and '新浪微博帐号' (Sina Weibo Account). The '微信帐号' option is currently selected. Below these options is a section for '人人网帐号' (Renren Account) with a login button and links to '用户服务协议' (User Service Agreement) and '隐私政策' (Privacy Policy).

On the right side of the Appium interface, the 'Recorder' tool is active, showing a list of Python code snippets for finding elements. The code is as follows:

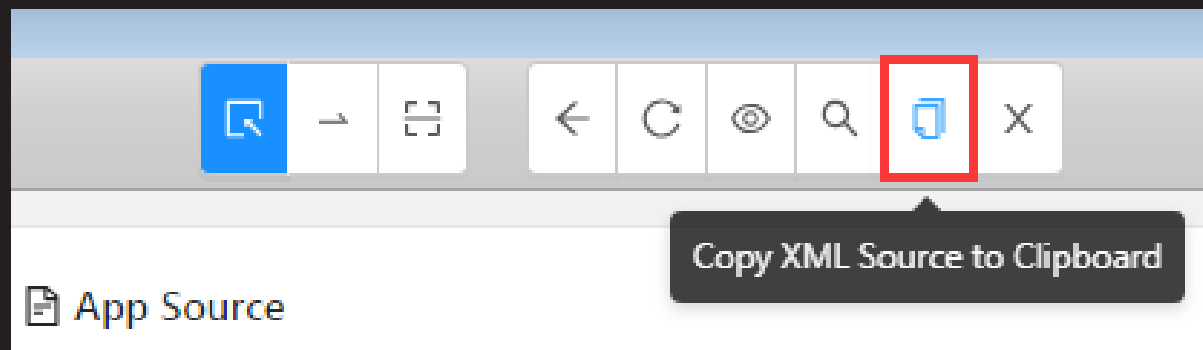
```

浏览器帐号")
els6 = driver.find_elements_by_name("搜狗浏览器帐号")
els7 = driver.find_elements_by_name("QQ帐号")
els8 = driver.find_elements_by_name("搜狗浏览器帐号")
els9 = driver.find_elements_by_name("搜狗浏览器帐号")
    
```

The line `els8 = driver.find_elements_by_name("搜狗浏览器帐号")` is highlighted with a red box. Below the code, the 'App Source' section shows the XML structure of the app's UI, including elements like `<android.widget.FrameLayout>`, `<android.widget.LinearLayout>`, and `<android.view.View resource-id="android:id/statusBarBackground">`.

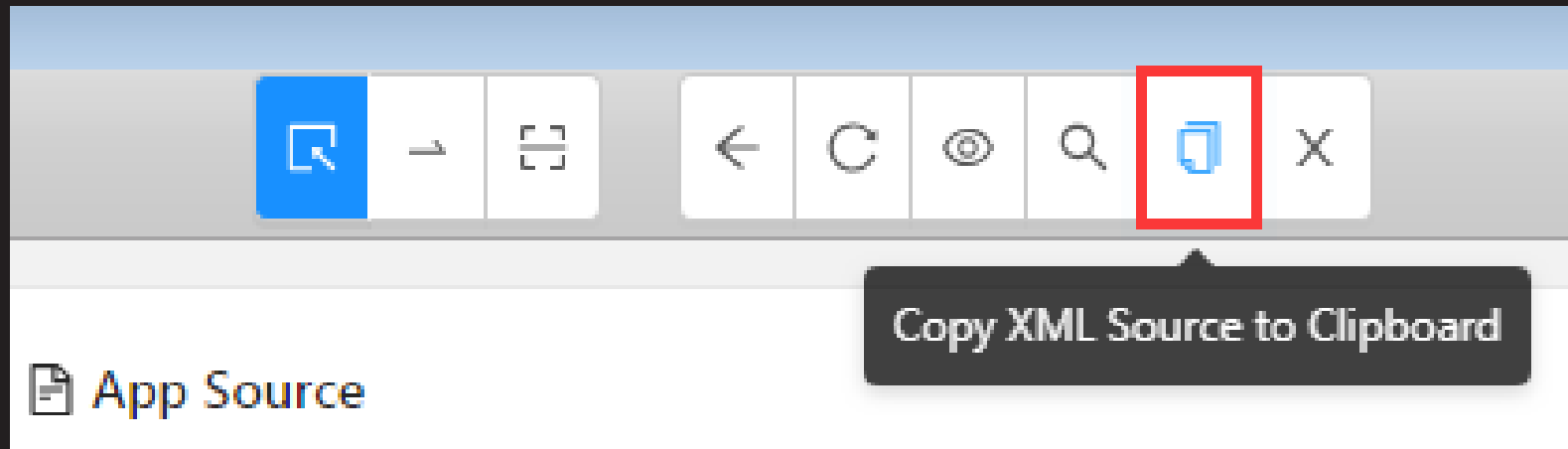
复制源码到剪切板

- 复制源码到剪切板 (Copy XML Source to Clipboard)



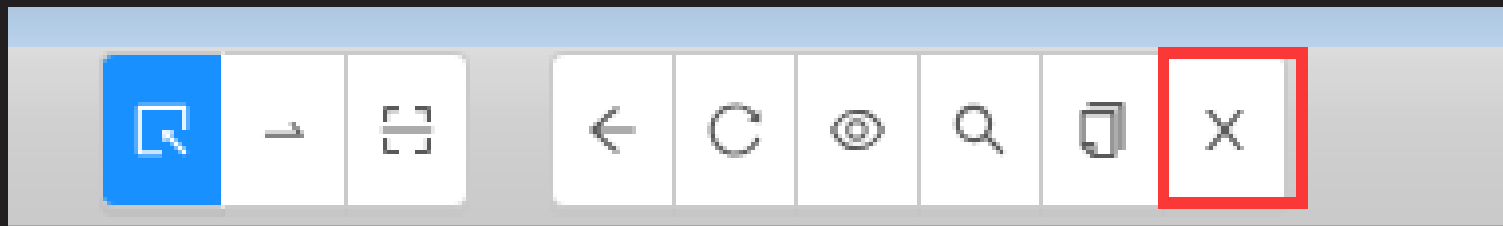
复制源码到剪切板

- 复制源码到剪切板 (Copy XML Source to Clipboard)



关闭会话

- 关闭会话 (Quit Session & Close Inspector)



录制通讯录App脚本

- 启动Inspector，录制App内操作步骤脚本，复制后，关闭会话，详见【COOKBOOK】



总结和答疑

