

# 目录

前言	1.1
简介	1.2
环境搭建	1.3
核心功能	1.4
监听	1.4.1
查找元素	1.4.2
举例：浏览器的输入框	1.4.2.1
等待元素出现	1.4.2.2
xpath	1.4.2.3
操作元素	1.4.3
点击元素	1.4.3.1
输入内容	1.4.3.2
获取元素属性	1.4.3.3
按键	1.4.4
屏幕	1.4.5
APP	1.4.6
其他功能	1.4.7
相关	1.5
weditor	1.5.1
adb	1.5.2
android-uiautomator-server	1.5.3
uiautomator	1.5.4
常见问题和经验	1.6
文字输入	1.6.1
xml源码不是最新的	1.6.2
重启服务	1.6.2.1
NAF	1.6.3
long_click不工作	1.6.4
后台服务已被杀掉	1.6.5
源码分析	1.7
常用代码段	1.8
工具类函数	1.8.1
adb相关	1.8.2

---

设备相关	1.8.3
其他	1.8.4
实际案例	1.9
百度搜索自动化	1.9.1
微信相关	1.9.2
确定类按钮	1.9.3
广告类弹框	1.9.4
Vivo自动安装app	1.9.5
Vivo账号自动登录	1.9.6
奇虎360自动登录账号	1.9.7
附录	1.10
参考资料	1.10.1

---

# 安卓自动化测试利器：uiautomator2

- 最新版本： v2.4
- 更新时间： 20210805

## 简介

总结安卓设备自动化测试领域好用的库uiautomator2，包括简介；如何搭建环境；有哪些核心功能，比如监听、用xpath或driver传参等方式查找元素，且给出了具体的浏览器的输入框如何查找定位的例子、以及常见的操作元素，比如点击元素、输入内容等、如何获取当前屏幕的截图和xml源码；以及与u2相关的内容，比如辅助调试的weditor、adb、android-uiautomator-server、uiautomator；以及常见问题和经验，比如文字输入、NAF、long\_click不工作、后台服务被杀掉等；以及一些源码分析；和通用代码段，包括工具类函数、adb相关、设备相关等；最后给出参考资料和文档。以及额外加上了很多实际案例，比如常见的确定类弹框按钮、自动关闭各大应用市场的广告类弹框、Vivo自动安装app、Vivo自动登录账号、奇虎360的自动登录账号。以及其他一些常见逻辑。

## 源码+浏览+下载

本书的各种源码、在线浏览地址、多种格式文件下载如下：

### Gitbook源码

- [crifan/android\\_automation\\_uiautomator2: 安卓自动化测试利器：uiautomator2](#)

如何使用此Gitbook源码去生成发布为电子书

详见：[crifan/gitbook\\_template: demo how to use crifan gitbook template and demo](#)

### 在线浏览

- [安卓自动化测试利器：uiautomator2 book.crifan.com](#)
- [安卓自动化测试利器：uiautomator2 crifan.github.io](#)

### 离线下载阅读

- [安卓自动化测试利器：uiautomator2 PDF](#)
- [安卓自动化测试利器：uiautomator2 ePUB](#)
- [安卓自动化测试利器：uiautomator2 Mobi](#)

### 版权说明

此电子书教程的全部内容，如无特别说明，均为本人原创和整理。其中部分内容参考自网络，均已备注了出处。如有发现侵犯您版权，请通过邮箱联系我 `admin 艾特 crifan.com`，我会尽快删除。谢谢合作。

## 鸣谢

感谢我的老婆陈雪的包容理解和悉心照料，才使得我 `crifan` 有更多精力去专注技术专研和整理归纳出这些电子书和技术教程，特此鸣谢。

## 更多其他电子书

本人 `crifan` 还写了其他 100+ 本电子书教程，感兴趣可移步至：

[crifan/crifan\\_ebook\\_readme: Crifan的电子书的使用说明](#)

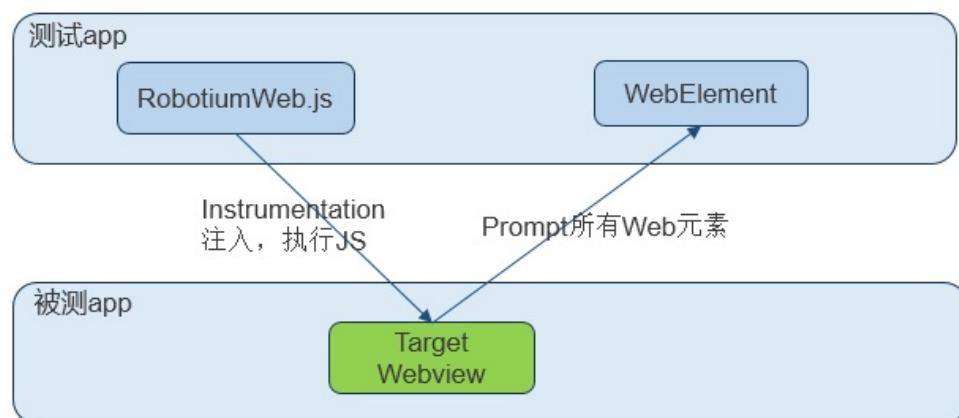
`crifan.com`, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2021-09-11 22:25:38

## uiautomator2简介

- uiAutomator2
  - 简称: u2
  - 是什么: 使用Python对Android设备进行UI自动化的库
  - 作用: 自动化操作安卓设备, 用于测试或抓包等
  - 语言: Python
  - 主页
    - [openatx/uiAutomator2: Android UiAutomator2 Python Wrapper](#)
    - 其中 openatx 中的
      - ATX = AutomatorX
  - 竞品=其他安卓自动化测试框架
    - Robotium
    - Selendroid
    - Espresso

## 基本原理

- 背景
  - Android内置的支持测试的框架
    - Android 4.2+: UiAutomator
    - Android 2.3 ~ 4.1: Instrumentation
- uiAutomator2的原理
  - 图



- 文字
  - 采用 Instrumentation 注入被测app后, 执行 js 脚本, 提取并封装成拥有 Web 元素的文本信息、 id 或 class 等属性、坐标信息等等的WebElement对象
  - 通过 js 注入的方式, 可以获取网页中的包括文字、tag标签、属性、坐标等等信息。
    - Android

- `WebChromeClient` 类在 Android 中，主要用于辅助 `WebView` 处理 js 的对话框、提示框等等

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2020-08-09 10:14:28

## 环境搭建

下面介绍如何搭建uiautomator2的开发环境，去测试安卓设备。

### 准备工作：安卓手机

#### 确保手机中开启了USB安装

安卓手机中开启 开发者选项 -> USB调试 -> USB安装



否则后续无法正常安装uiautomator2相关的ATX等软件和服务。

**安装**

```
pip3 install -U uiautomator2
```

- 如果包管理器是 pipenv，则用：

```
pipenv install uiautomator2
```

- 最新版( 1.3.0+ )无需再去 python3 -m uiautomator2 init
  - 后续代码首次 u2.connect() 会自动安装所需的apk的

## 测试连接

再去测试连接：

```
import uiautomator2 as u2
d = u2.connect() # connect to device
print(d.info)
```

其中：

u2.connect() 可以换成 wifi 或 usb：

- wifi

```
d = u2.connect('10.0.0.1')
```

- usb

```
d = u2.connect('8c8a4d4d')
```

- 其中 8c8a4d4d 是 adb devices 列出的当前（用USB数据线连接到Mac中的）安卓设备的ID

```
→ ~ adb devices
List of devices attached
8c8a4d4d    device
```

输出举例：

```
→ autoTestAndroidGameHappyBigBattle python
Python 3.7.3 (default, May 22 2019, 10:55:14)
[Clang 10.0.1 (clang-1001.0.46.4)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import uiautomator2 as u2
>>> d = u2.connect('8c8a4d4d')
conn <urllib3.connection.HTTPConnection object at 0x1077f4da0 ,method GET,url /version,tim
eout_obj Timeout(connect 2, read 2, total None),body None,headers {'User-Agent': 'python-req
uests/2.22.0', 'Accept-Encoding': 'gzip, deflate', 'Accept': '*/*', 'Connection': 'keep-
alive'},chunked False
```

和：红米Note10X打开了微信的输出：

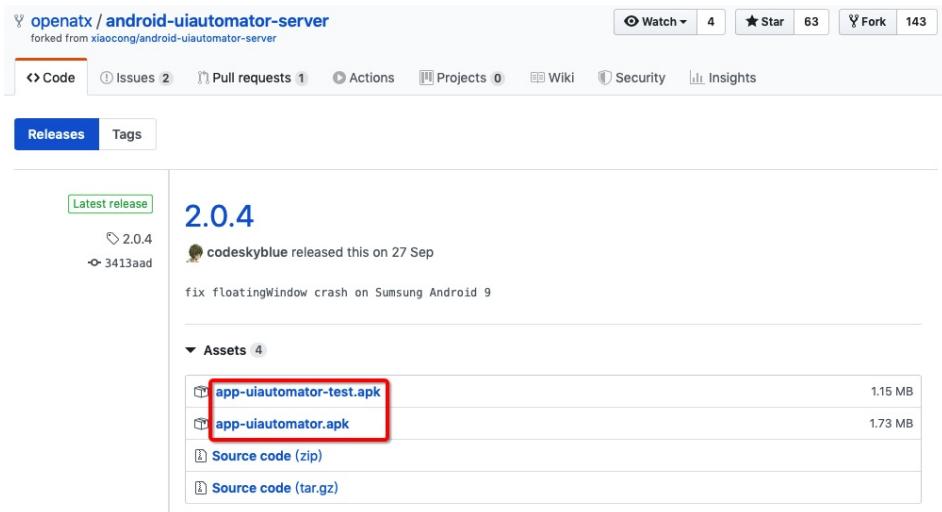
```
{'currentPackageName': 'com.tencent.mm', 'displayHeight': 2201, 'displayRotation': 0, 'displaySizeDpX': 393, 'displaySizeDpY': 873, 'displayWidth': 1080, 'productName': 'atom', 'screenOn': True, 'sdkInt': 29, 'naturalOrientation': True}
```

## 说明：安装细节

### 安装内容

上述命令会安装相关工具到你安卓手机中：

- uiautomator-server
  - 作用：包含http rpc服务的apk
    - 2个apk
    - 图解



- 框架要求2个apk，缺一不可
  - app-uiautomator-test.apk：测试程序
    - uiautomator这个框架允许我们测试第三方应用
    - 包名：com.github.uiautomator.test
  - app-uiautomator.apk：被测应用
    - 基本就是个傀儡
      - 只要别轻易的死掉，就算是一个合格的应用了
      - 包名：com.github.uiautomator
  - 地址：<https://github.com/openatx/android-uiautomator-server/releases>
- atx-agent
  - 地址：<https://github.com/openatx/atx-agent>
- openstf/minicap
  - 地址：<https://github.com/openstf/minicap>
- openstf/minitouch
  - 地址：<https://github.com/openstf/minitouch>

## 安装log日志

期间如果开启了uiautomator2的debug后，可以看到更详细的信息。

比如安装路径（小米9中安装期间显示安装的东西有）：

- minicap、minitouch
  - [https://tool.appetizer.io/openatx/stf-binaries/raw/master/node\\_modules/minitouch-prebuilt/prebuilt/arm64-v8a/bin/minitouch](https://tool.appetizer.io/openatx/stf-binaries/raw/master/node_modules/minitouch-prebuilt/prebuilt/arm64-v8a/bin/minitouch)
- com.github.uiautomator, com.github.uiautomator.test 2.0.3
  - <https://tool.appetizer.io/openatx/android-uiautomator-jsonrpcserver/releases/download/v0.1.6/bundle.jar>
  - <https://tool.appetizer.io/openatx/android-uiautomator-jsonrpcserver/releases/download/v0.1.6/uiautomator-stub.jar>
  - <https://tool.appetizer.io/openatx/android-uiautomator-server/releases/download/2.0.3/app-uiautomator.apk>
  - <https://tool.appetizer.io/openatx/android-uiautomator-server/releases/download/2.0.3/app-uiautomator-test.apk>

安卓6的 华为畅享6S，重新初始化的log是：

```
[200218 13:55:44] [DevicesMethods.py 11 ] start init driver
[I 200218 13:55:45 init:132] uiautomator2 version: 2.5.3
[I 200218 13:55:45 init:317] Install minicap, minitouch
[I 200218 13:55:45 init:330] Install com.github.uiautomator, com.github.uiautomator.test 2
.1.1
[I 200218 13:56:02 init:300] - app-uiautomator.apk installed
[I 200218 13:56:14 init:300] - app-uiautomator-test.apk installed
[I 200218 13:56:14 init:308] Install atx-agent 0.8.2
[I 200218 13:56:19 init:342] Check atx-agent version
Successfully init AdbDevice(serial DWH9X17124W03779)
```

安卓9的 红米Note8Pro 的初始化log是：

```
[200217 14:45:33] [DevicesMethods.py 11 ] start init driver
[I 200217 14:45:37 init:132] uiautomator2 version: 2.5.3
[I 200217 14:45:37 init:317] Install minicap, minitouch
minicap.so |       | 67.1K/67.1K
[I 200217 14:45:37 init:330] Install com.github.uiautomator, com.github.uiautomator.test 2
.1.1
[I 200217 14:45:38 init:300] - app-uiautomator.apk installed
[I 200217 14:45:38 init:300] - app-uiautomator-test.apk installed
[I 200217 14:45:38 init:308] Install atx-agent 0.8.2
[I 200217 14:45:39 init:342] Check atx-agent version
Successfully init AdbDevice(serial hmucae175ptk7szs)
```

分别对应着去安装：

- minicap和minitouch

- com.github.uiautomator和com.github.uiautomator.test
  - 对应着：app-uiautomator.apk和app-uiautomator-test.apk
- atx-agent

## 安装后的app

不过，实际上（安卓10的小米9，安卓9的小米Note8Pro）只安装了，最核心的2个：

- ATX
  - 桌面图标



- 安装期间需要手动点击 继续安装



- com.github.uiautomator.test
  - 桌面图片：无
  - 安装期间，需要手动点击：继续安装



安装后，可以在应用管理中找到，刚才安装的2个应用：

- 红米Note8Pro 安卓9

- - 华为畅享6S 安卓6

◦

## ATX

关于ATX，启动后的主界面：



点击 启动UIAUTOMATOR 后，会显示： ATX: Uiautomator started



crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2021-08-05 20:09:22

## 核心功能

接着介绍uiautomator2的一些常用的核心的功能。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09 10:14:28

## 监听

其中一个很常用的功能就是：监听

即，注册了要监听的条件，满足后，就会自动触发。

典型应用比如，希望界面中出现 好的、确定 等按钮，就自动点击。

比如：

- 大众点评安装期间的 安装

◦

- 弹框中的 允许

- - 普通安卓原生的按钮：确定

◦

则需要去注册监听器，其核心逻辑是：

- 之前用: `watcher`
- 后改用: `xpath`
- 20210239 作者后来又改回: `watcher`

详细解释:

## 用 `watcher` 实现监听

```
# 注册单个监听器
d.watcher("安装").when(text="安装").click()
# 等价于
d.watcher("安装").when(text="安装").click(text="安装")

# (此刻) 单次运行 (一次)
d.watchers.run()

# 后台长期的运行
# d.watchers.watched = True # 旧
d.driver.watcher.start() # 新
```

其中的:

- 20210329更新: 版本 v2.5.3 之后, 又从 `xpath` 换回 `watcher`
  - 之前: `d.watchers.watched` 在 `uiautomator2 >=1.0.0` 版本后已废弃。
  - 当时: 推荐换用下面的 `xpath` 的写法: `xpath.watch_background`

实际调用举例:

```
def register_watcher(self):
    # Note: since uiAutomator2 v2.5.3, change xpath back to wather, so use watcher now
    for key in self.config["install"]:
        logging.debug("register {}".format(key))
        # self.driver.watcher(key).when(text=key).click()
        # self.driver.watcher(key).when(text=key).click(text=key)
        # self.driver.watcher.when(key).click()
        self.driver.watcher(key).when(key).click()

    XpathConfigKeyList = [
        "Confirm_Button_Xpath_List",
        "NextStep_Button_Xpath_List",
        "PopupWindow_CloseButton_Xpath_List",
    ]
    for eachXpathConfigKey in XpathConfigKeyList:
        curXpathList = self.config[eachXpathConfigKey]
        for eachXpath in curXpathList:
            self.driver.watcher.when(eachXpath).click()
            logging.debug("Registered xpath wathcher: %s", eachXpath)

    self.driver.watcher.when(self.config["Vivo_Password_Input_Xpath"]).call(self.autoInputVivoPassword)
```

```

    self.driver.watcher.when(self.config["Vivo_Register_Vivo_Account_Xpath"]).call(self.auto
DoVivoAccountLogin)
    self.driver.watcher.when(self.config["Permission_Settings_GoTo_App_Config_Xpath"]).call(
self.autoGrantPermission)
    # self.driver.watcher.when(self.config["Qihoo360_Login_ReadAndAgree_Xpath"]).call(self.a
utoDo360AccountLogin)
    self.driver.watcher.when(self.config["Qihoo360_PasswordLogin_Xpath"]).call(self.autoDo36
0AccountLogin)

    self.driver.watcher.start()

```

## 用 xpath 实现监听

```

# 注册单个监听器
d.xpath.when(text "安装").click()

# 单次运行一次
d.xpath.run_watchers()

# 后台长期的运行=开启后台监控模式
d.xpath.watch_background() # 默认每4s检查一次
# 或手动设置间隔时间
d.xpath.watch_background(2.0) # 2.0表示每2秒检查一次

# 如果需要，再去停止后台监听
d.xpath.watch_stop()

```

更多关于xpath的细节和用法，详见：

[uiautomator2/uiautomator2/ext>xpath at master · openatx/uiautomator2](#)

(注：不在主页的readme中，所以一般很少人能找到。我是从[raw的readme.md](#)中反推才找到的)

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2021-07-18 09:59:23

## 查找元素

安卓测试期间，最常用的要属于，查找和定位页面中的相关元素了。

目前主要有2种方式去查找(定位) 元素：

- 调用 `driver`，直接传递(属性值对应的)参数
  - 返回值类型：`uiautomator2.session.UiObject`
    - 获取元素属性方式：`UiObject.info`
- 调用 `driver.xpath` 函数，传入xpath路径
  - 返回值类型：`uiautomator2.xpath.XPathSelector`
    - 获取元素属性方式：
      - 先：`XPathSelector.get()` 得到 `uiautomator2.xpath.XMLElement`
      - 再：`XMLElement.attrib` 才能获取属性值
    - 注：
      - `xpath(xpathSelecotr).all()` 得到是 `XMLElement` 的 `list` 列表，而不是 `XPathSelector`
        - -> `XPathSelector` 去 `get()` 或 `all()` 后，都是 `XMLElement`

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2021-07-18 09:59:23

## 举例：定位小米手机中内置浏览器中的输入框和获取属性值

下面举例来介绍：

### 页面

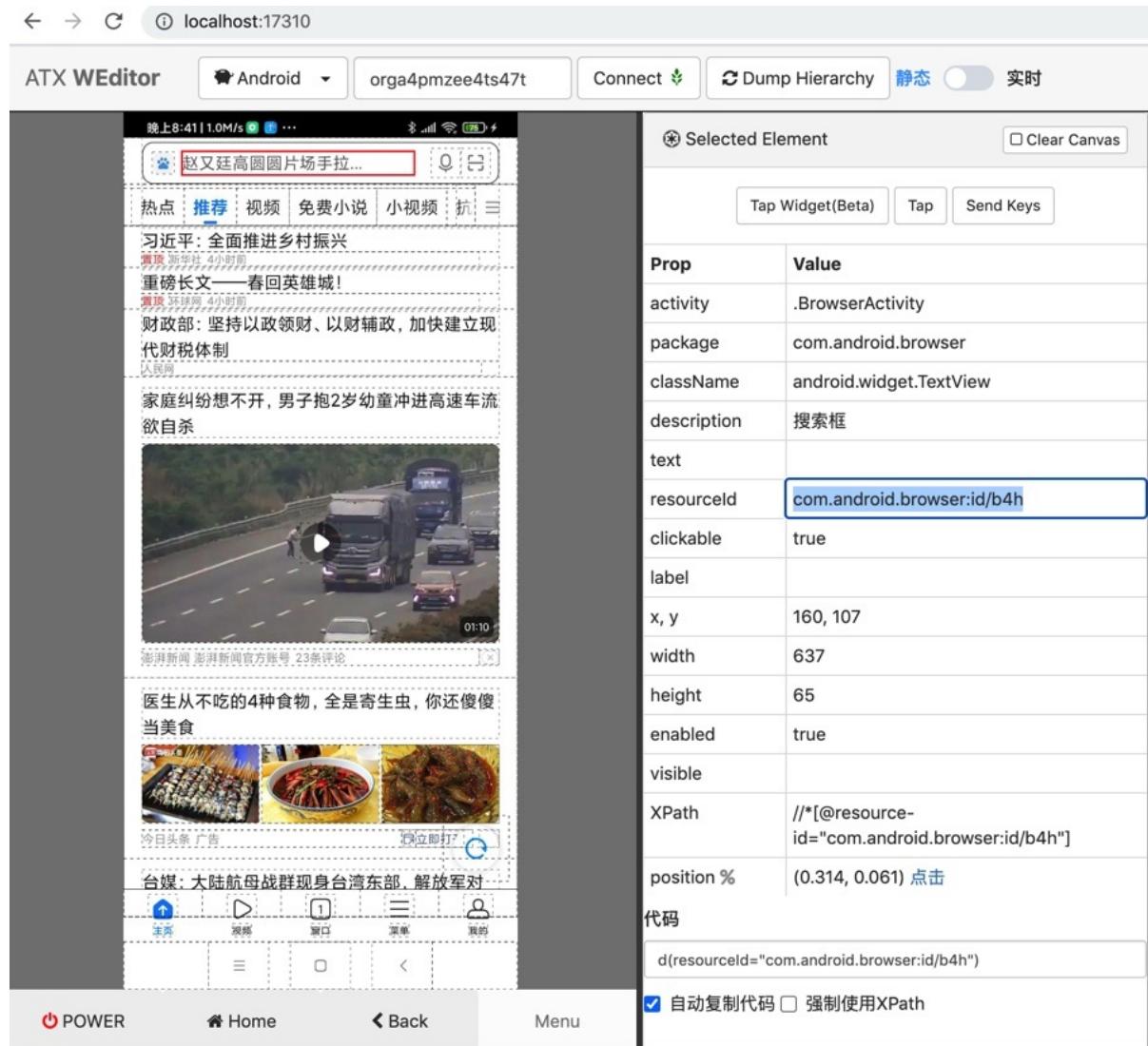
对于安卓手机 小米Note10X 其中内置的浏览器的首页中顶部的输入框：



## 搞清楚元素属性

通过多种调试：

借助 `weditor` 调试页面，看元素属性



拷贝出全部属性：

```
Selected Element

Prop      Value
activity     .BrowserActivity
package      com.android.browser
className    android.widget.TextView
description  搜索框
text
resourceId   com.android.browser:id/b4h
clickable    true
label
x, y        160, 107
width       637
height      65
enabled     true
visible
XPath       //*[@@resource-id = "com.android.browser:id/b4h"]
position    (0.314, 0.061) 点击
```

```
代码 d(resourceId="com.android.browser:id/b4h")
```

用代码 `dump_hierarchy` 输出整页的xml中找到对应元素的部分

```
curPageXml = d.dump_hierarchy(compressed=False, pretty=False)
```

输出的页面xml源码中，有对应该元素的部分

```
<node index="1" text="" resource-id="com.android.browser:id/b4h" class="android.widget.TextView" package="com.android.browser" content-desc="搜索框" checkable="false" checked="false" clickable="true" enabled="true" focusable="false" focused="false" scrollable="false" long-clickable="false" password="false" selected="false" visible-to-user="true" bounds="[160,107][797,172]" />
```

都可得知对应属性是：

- package : com.android.browser
- class : android.widget.TextView
- resource-id : com.android.browser:id/b4h
- content-desc : 搜索框
- ...

## 查找（定位）元素

此处去查找定位该元素的逻辑就是：

- `class="android.widget.TextView" + resource-id="com.android.browser:id/b4h"`
  - 注：如果确定 resource id 的值是唯一的，则也可以只用：
    - `resource-id="com.android.browser:id/b4h"`

对应代码是：

用 `driver(propertyParameter)` 去查找元素

```
inputUiObj = d(resourceId="com.android.browser:id/b4h", className="android.widget.TextView")
```

或

```
inputUiObj = d(resourceId="com.android.browser:id/b4h")
```

注：其他属性对应的参数还有：

- `text`, `textContains`, `textMatches`, `textStartsWith`
- `className`, `classNameMatches`
- `description`, `descriptionContains`, `descriptionMatches`, `descriptionStartsWith`
- `checkable`, `checked`, `clickable`, `longClickable`
- `scrollable`, `enabled`, `focusable`, `focused`, `selected`
- `packageName`, `packageNameMatches`
- `resourceId`, `resourceIdMatches`
- `index`, `instance`

去找到元素。

关于返回元素的类型，可以通过

```
print("type(inputUiObj)=%s" % type(inputUiObj))
```

得知是：

```
<class 'uiautomator2.session.UiObject'>
```

想要继续获取元素属性，则可以通过 `info`

```
inputUiObjectInfo = inputUiObj.info
```

关于 `info` 的类型，可以通过

```
print("type(inputUiObjectInfo)=%s" % type(inputUiObjectInfo))
```

得知是：

```
<class 'dict'>
```

打印出的 `info` 的值是：

```
inputUiObjectInfo={'bounds': {'bottom': 172, 'left': 160, 'right': 797, 'top': 107}, 'childCount': 0, 'className': 'android.widget.TextView', 'contentDescription': '搜索框', 'packageName': 'com.android.browser', 'resourceName': 'com.android.browser:id/b4h', 'text': '', 'visibleBounds': {'bottom': 172, 'left': 160, 'right': 797, 'top': 107}, 'checkable': False, 'checked': False, 'clickable': True, 'enabled': True, 'focusable': False, 'focused': False, 'longClickable': False, 'scrollable': False, 'selected': False}
```

附上完整代码：

```
# Method 1: use driver pass in parameter
```

```
inputUiObj = d(resourceId="com.android.browser:id/b4h", className="android.widget.TextView")
# inputUiObj = d(resourceId="com.android.browser:id/b4h")
print("type(inputUiObj)=%s" % type(inputUiObj)) # type(inputUiObj)=<class 'uiautomator2.session.UiObject'>
print("inputUiObj=%s" % inputUiObj) # inputUiObj=<uiautomator2.session.UiObject object at 0x10a0bea00>
inputUiObjectInfo = inputUiObj.info
print("type(inputUiObjectInfo)=%s" % type(inputUiObjectInfo)) # type(inputUiObjectInfo)=<class 'dict'>
print("inputUiObjectInfo=%s" % inputUiObjectInfo) # inputUiObjectInfo={'bounds': {'bottom': 172, 'left': 160, 'right': 797, 'top': 107}, 'childCount': 0, 'className': 'android.widget.TextView', 'contentDescription': '搜索框', 'packageName': 'com.android.browser', 'resourceName': 'com.android.browser:id/b4h', 'text': '', 'visibleBounds': {'bottom': 172, 'left': 160, 'right': 797, 'top': 107}, 'checkable': False, 'checked': False, 'clickable': True, 'enabled': True, 'focusable': False, 'focused': False, 'longClickable': False, 'scrollable': False, 'selected': False}
isFoundInput = inputUiObj.exists # True
```

## 用 `driver.xpath(xpathSelector)` 去查找元素

```
inputXPathSelector = d.xpath("//android.widget.TextView[@resource-id='com.android.browser:id/b4h'])")
```

或：

```
inputXPathSelector = d.xpath("//*[@resource-id='com.android.browser:id/b4h'])")
```

找到元素。

关于 `xpath` 返回的类型，可以通过

```
print("type(inputXPathSelector)=%s" % type(inputXPathSelector))
```

得知是：

```
<class 'uiautomator2.xpath.XPathSelector'>
```

想要获取属性值，此处要先去得到 `XPathSelector` 对应的XML元素

```
inputXPathElem = inputXPathSelector.get()
```

关于 `get()` 返回的类型，可以通过

```
print("type(inputXPathElem)=%s" % type(inputXPathElem))
```

得知是：

```
<class 'uiautomator2.xpath.XMLElement'>
```

然后才能去用 `inputXPathElem.attrib` 获取属性值

关于 `attrib` 的类型，可以通过

```
print("type(inputXPathElem.attrib)=%s" % type(inputXPathElem.attrib))
```

得知是：

```
<class 'lxml.etree._Attrib'>
```

打印出的 `dict` 类型的 `inputXPathElem.attrib` 值是：

```
inputXPathElem.attrib={'index': '1', 'text': '', 'resource-id': 'com.android.browser:id/b4h', 'package': 'com.android.browser', 'content-desc': '搜索框', 'checkable': 'false', 'checked': 'false', 'clickable': 'true', 'enabled': 'true', 'focusable': 'false', 'focused': 'false', 'scrollable': 'false', 'long-clickable': 'false', 'password': 'false', 'selected': 'false', 'visible-to-user': 'true', 'bounds': '[160,107][797,172]'}
```

附上完整代码：

```
# Method 2: use xpath
inputXPathSelector = d.xpath("//android.widget.TextView[@resource-id='com.android.browser:id/b4h']")
# inputXPathSelector = d.xpath("//*[@resource-id='com.android.browser:id/b4h']")
print("type(inputXPathSelector)=%s" % type(inputXPathSelector)) # type(inputXPathSelector)=<class 'uiautomator2.xpath.XPathSelector'>
inputXPathElem = inputXPathSelector.get()
print("type(inputXPathElem)=%s" % type(inputXPathElem)) # type(inputXPathElem)=<class 'uiautomator2.xpath.XMLElement'>
print("inputXPathElem=%s" % inputXPathElem) # inputXPathElem=<uiautomator2.xpath.XMLElement object at 0x108585d30>
print("type(inputXPathElem.attrib)=%s" % type(inputXPathElem.attrib)) # type(inputXPathElem.attrib)=<class 'lxml.etree._Attrib'>
print("inputXPathElem.attrib=%s" % inputXPathElem.attrib) # inputXPathElem.attrib={'index': '1', 'text': '', 'resource-id': 'com.android.browser:id/b4h', 'package': 'com.android.browser', 'content-desc': '搜索框', 'checkable': 'false', 'checked': 'false', 'clickable': 'true', 'enabled': 'true', 'focusable': 'false', 'focused': 'false', 'scrollable': 'false', 'long-clickable': 'false', 'password': 'false', 'selected': 'false', 'visible-to-user': 'true', 'bounds': '[160,107][797,172]'}
isFoundInput = inputXPathSelector.exists # True
```



## 等待元素出现

当想要实现，（点击搜索按钮等操作后）希望某个元素出现，表示页面加载成功，可以用： `exists`  
举例：

```
Browser_XiaomiBuiltin = "com.android.browser"
browserPackage = Browser_XiaomiBuiltin
# wait util browser launch complete -> appear 主页 tab
d(text "主页", packageName=packageName).exists(timeout 10)
```

可以实现： 等待足够长时间（最长10秒），直到出现

- package是"com.android.browser"
  - 小米的内置浏览器
- text是"主页"

表示浏览器的左下角第一个tab

以此表示 浏览器启动完毕了

从而实现了，等待（浏览器启动和加载）完毕（直到出现某个页面上的元素）的效果

后记：后来发现，左下角第一个tab，有时候会变，比如变成了： 资讯

所以去用最后下角的 我的 tab，这个不会变

代码改为：

```
# wait util browser launch complete -> appear 我的 tab
# MustShowTabName = "主页"
MustShowTabName = "我的"
# d(text=MustShowTabName).exists(timeout=10)
d(text MustShowTabName, packageName=packageName).exists(timeout 10)
```

即可。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2021-07-18 09:59:23

## xpath

xpath本身是一套独立的技术，常用于web领域内。

此处uiautomator2也支持xpath，用于元素定位，可以实现复杂条件的元素的查找。

### xpath实际用法举例

想要找到百度搜索结果的results下面的button，则对应语法是：

```
titleButtonSelectorList = d.xpath("//*[@resource-id='results']//android.widget.Button[@clickable='true']").all()
```

即可找到对应元素的列表

```
searchStr: 'crifan'
> searchUiObj: <uiAutomator2.session.UiObject object at 0x108...
> time: <module 'time' (built-in)
titleButtonNum: 3
< titleButtonSelectorList: [<uiAutomator2.xpath...108873040>...
> special variables
> function variables
> 0: <uiAutomator2.xpath.XMLElement object at 0x108873040>
> 1: <uiAutomator2.xpath.XMLElement object at 0x108873250>
> 2: <uiAutomator2.xpath.XMLElement object at 0x1088737f0>
len(): 3
135
136
137
138
139
140
141
142
143
144
145
146
147
```

注意：

分开写

```
resultsSelector = d.xpath("//*[@resource-id='results']")
titleButtonSelectorList = resultsSelector.xpath("//android.widget.Button[@clickable='true']").all()
```

却就是找不到元素。

具体原因，暂时未知。

### xpath常见操作

#### 定位节点和操作节点

```
tbsNodeList = self.driver.xpath("//com.tencent.tbs.core.webkit.WebView").all()
```

xpath的all()返回的是XMLElement而不是XPathSelector

```
self.driver.xpath("//com.tencent.tbs.core.webkit.WebView") 返回的
是 uiautomator2.xpath.XPathSelector
```

而加上了 `all()` 后

```
self.driver.xpath("//com.tencent.tbs.core.webkit.WebView").all()
```

则返回的是： `uiautomator2.xpath.XMLElement` 的 list 了

即，每一个都是 `uiautomator2.xpath.XMLElement`，而不是 `uiautomator2.xpath.XPathSelector`

其中每一个都是 `uiautomator2.xpath.XMLElement`

然后就可以去操作属性了：

## 获取属性content-desc的值

```
eachXmlElem.attrib.get("content-desc", "")
```

## 给属性content-desc设置值

```
eachXmlElem.attrib["content-desc"] = "add something to avoid NAF"
```

## 删除一个属性

```
eachXmlElem.attrib.pop("NAF")
```

## 文档

关于Xpath的详细用法，见官网中的：

- xpath的文档
  - 新：[uiautomator2/XPATH.md at master · openatx/uiautomator2](#)
  - 旧：[uiautomator2/uiautomator2/ext/xpath at master · openatx/uiautomator2](#)

其内部用的lxml，具体功能和语法都可以参考：

[The lxml.etree Tutorial](#)

## 查找元素 相关函数

`findAndClickNode`： 查找当前节点的父级符合条件的节点 并点击

```
def findAndClickNode(self, curNodeXPath):
    """
    寻找可以clickable=true的当前或父级元素，并点击

    注：主要用于当节点clickable=false，点击无效时，使用此方法
    """

    # 找到所有父级元素
    parentElements = self.driver.xpath(curNodeXPath).parent
```

```

"""
foundAndClicked = False
matchDict = {"clickable": "true"}
clickableParentNode = self.findParentNode(curNodeXpath, curNodeXpath, matchDict)
if clickableParentNode:
    foundNodeAttrib = clickableParentNode.attrib
    clickableParentNode.click()
    foundAndClicked = True
    logging.info("clicked element [%s] found by [xpath=%s, match=%s]", foundNodeAttrib,
curNodeXpath, matchDict)
else:
    logging.warning("Fail click %s for not found %s(parent) node", curNodeXpath, matchDict)

return foundAndClicked

```

调用：

```

if curNodeXpath:
    foundAndClicked = self.findAndClickNode(curNodeXpath)

```

相关函数：

findParentNode：寻找父节点

```

def findParentNode(self, curNodeXpath, matchDict, maxUpLevel=3):
"""
    寻找符合特定条件的父级节点，最多向上找3级

    如果当前节点符合条件，则返回当前节点
"""

matchNode = None

try:
    curNode = self.driver.xpath(curNodeXpath).get()
    curNodeAttrib = curNode.attrib # .attrib contain 'clickable'
    # curNodeInfo = curNode.info # .info not contain 'clickable'
    isCurMatch = self.isMatchNode(curNodeAttrib, matchDict)
    if isCurMatch:
        # current is match
        matchNode = curNode
    else:
        # try parent nodes
        curUpLevel = 1
        curParentNodeXpath = curNodeXpath
        while(curUpLevel <= maxUpLevel):
            curParentNodeXpath += "/.."
            curParentNode = self.driver.xpath(curParentNodeXpath).get()
            curParentNodeAttrib = curParentNode.attrib

```

```

        isCurParentMatch = self.isMatchNode(curparentNodeAttrib, matchDict)
        if isCurParentMatch:
            matchNode = curparentNode
            break

        curUpLevel += 1

    except XPathElementNotFoundError as xpathNotFoundErr:
        logging.error("XPathElementNotFoundError: %s", xpathNotFoundErr)

    if not matchNode:
        logging.warning("Not found match parent for xpath=%s and match=%s", curNodeXpath,
matchDict)

    return matchNode

```

isMatchNode: 节点是否匹配

```

def isMatchNode(self, curNodeAttrib, toMatchInfo):
    """判断当前节点属性是否满足条件"""
    isAllMatch = True
    for eachKey, eachToMatchValue in toMatchInfo.items():
        if eachKey not in curNodeAttrib:
            isAllMatch = False
            break

        curValue = curNodeAttrib[eachKey]
        if curValue != eachToMatchValue:
            isAllMatch = False
            break

    return isAllMatch

```

findAndClickTextNode: 寻找节点并点击

```

def findAndClickTextNode(self, text):
    """
    对于text类型节点: android.widget.TextView, text=xxx
    寻找可以clickable=true的当前或父级元素，并点击

    注: 主要用于当text=xxx的节点clickable=false, 点击无效时, 使用此方法
    """
    curTextNodeXpath = "//android.widget.TextView[@text='%s']" % text
    self.findAndClickNode(curTextNodeXpath)

```

xpathFindElement: 用xpath查找元素

```

def xpathFindElement(self, curClass=None, curId=None, curBounds=None):

```

```

"""
    find element by xpath

    return value type
        is: u2.xpath.XMLElement
        not: u2.session.UiObject
"""

foundElement = None
curXPath = self.generateElementXPath(curClass curClass, curId curId, curBounds curBounds
ds)
try:
    foundElement = self.driver.xpath(curXPath).get()
except XPathElementNotFoundError as xpathNotFoundErr:
    logging.error("XPathElementNotFoundError: %s from %s", xpathNotFoundErr, curXPath)

return foundElement

```

调用：

(1)

```
foundElement = self.xpathFindElement(curClass locatorClass, curId locatorId, curBounds locatorBounds)
```

相关函数：

generateElementXPath：生成元素xpath

```

def generateElementXPath(self, curClass=None, curId=None, curBounds=None):
    """generate element xpath"""
    # nodeXPath = ""
    # if locatorClass:
    #     nodeXPath = "//%s[@bounds='%s']" % (locatorClass, locatorBounds) # "//android.widget.TextView[@bounds='[191,2060][430,2135]']"
    # elif locatorId:
    #     nodeXPath = "//*[@resource-id='%s' and @bounds='%s']" % (locatorId, locatorBounds)
    # else:
    #     nodeXPath = "//*[@bounds='%s'" % locatorBounds

    classRule = "*"
    if curClass:
        classRule = curClass # 'android.widget.ImageView'

    propertyRule = ""
    if curId:
        propertyRule += "@resource-id='%s'" % curId
        # "@resource-id='com.netease.newsreader.activity:id/hs'"

    if curBounds:
        if propertyRule:

```

```

        propertyRule += " and "

    propertyRule += "@bounds='%s'" % curBounds
    # "@resource-id='com.netease.newsreader.activity:id/hs' and @bounds='[75,2098][141,2134]'

    # TODO: add other support: text, desc, instance, ...
    curXPath = "//%s[%s]" % (classRule, propertyRule)
    # "//android.widget.ImageView[@resource-id='com.netease.newsreader.activity:id/hs' and
    # @bounds='[75,2098][141,2134]']"

    return curXPath

```

调用:

```

curClassname = None
curResId = None
curBoundsStr = None

# curAttrib = foundElement.attrib
# AttributeError: 'UiObject' object has no attribute 'attrib'
if hasattr(foundElement, "attrib"):
    curAttrib = foundElement.attrib
    # {'index': '0', 'text': '', 'resource-id': 'com.netease.newsreader.activity:id/hs', 'package': 'com.netease.newsreader.activity', 'content-desc': '', 'checkable': 'false', 'checked': 'false', 'clickable': 'false', 'enabled': 'true', 'focusable': 'false', 'focused': 'false', 'scrollable': 'false', 'long-clickable': 'false', 'password': 'false', 'selected': 'true', 'visible-to-user': 'true', 'bounds': '[75,2098][141,2134]'}
    curResId = curAttrib["resource-id"]
    curBoundsStr = curAttrib["bounds"]
else:
    # # for debug
    # self.debugPrintElement(foundElement, "no attrib")
    logging.debug("")

curInfo = foundElement.info
# {'bounds': {'bottom': 2134, 'left': 75, 'right': 141, 'top': 2098}, 'className': 'android.widget.ImageView', 'contentDescription': '', 'enabled': 'true', 'focusable': 'false', 'focused': 'false', 'longClickable': 'false', 'packageName': 'com.netease.newsreader.activity', 'scrollable': 'false', 'selected': 'true', 'text': ''}

if not curClassname:
    curClassname = curInfo["className"] # 'android.widget.ImageView'

if not curBoundsStr:
    boundsDict = curInfo["bounds"]
    x0 = boundsDict["left"]
    y0 = boundsDict["top"]
    x1 = boundsDict["right"]
    y1 = boundsDict["bottom"]
    curBoundsStr = "[%d,%d][%d,%d]" % (x0, y0, x1, y1)
    # '[75,2098][141,2134]'

```

```

if not curResId:
    if "resourceName" in curInfo:
        curResId = curInfo["resourceName"] # 'com.netease.newsreader.activity:id/bn5'

curNodeXPath = self.generateElementXPath(
    curClass curClassName,
    curId curResId,
    curBounds curBoundsStr,
)

```

## 查找元素

```

def find_element_Android(self, locator):
    """Android: find element"""
    foundElement = None
    locatorType = locator.get("type")
    locatorText = locator.get("text")
    locatorClass = locator.get("class")
    locatorDesc = locator.get("desc")
    locatorId = locator.get("id")
    locatorInstance = locator.get("instance")
    locatorBounds = locator.get("bounds")

    if locatorType:
        if locatorType == "text":
            foundElement = self.driver(text=locatorText)
        elif locatorType == "desc":
            foundElement = self.driver(description=locatorDesc)
        elif locatorType == "id":
            foundElement = self.driver(resourceId=locatorId)
        elif locatorType == "id+bounds":
            foundElement = self.xpathFindElement(curClass=locatorClass, curId=locatorId
, curBounds=locatorBounds)
        elif locatorType == "class+bounds":
            foundElement = self.xpathFindElement(curClass=locatorClass, curBounds=locatorBounds)
        elif locatorType == "id+text":
            foundElement = self.driver(resourceId=locatorId, text=locatorText)
        elif locatorType == "id+desc":
            foundElement = self.driver(resourceId=locatorId, description=locatorDesc)
        elif locatorType == "class+instance":
            # foundElement = self.driver(className=locator["class"], instance=locatorI
nstance)
            instanceInt = int(locatorInstance)
            # foundElement = self.driver(className=locatorClass, instance=instanceInt)
            foundElementList = self.driver(className=locatorClass)
            if foundElementList:
                curIdx = instanceInt
                shouldMaxNumber = curIdx + 1
                if foundElementList.count >= shouldMaxNumber:
                    foundElement = foundElementList[curIdx]

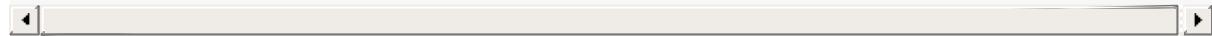
```

```

if not foundElement:
    if locatorText:
        foundElement = self.driver(text=locatorText)
    elif locatorClass:
        foundElement = self.driver(className=locatorClass)
    elif locatorDesc:
        foundElement = self.driver(description=locatorDesc)
    elif locatorId:
        foundElement = self.driver(resourceId=locatorId)
    elif locatorInstance:
        foundElement = self.driver(instance=locatorInstance)
    elif locatorBounds:
        foundElement = self.xpathFindElement(curBounds=locatorBounds)

return foundElement

```



调用：

```

EditTextClass = "android.widget.EditText"

curBounds = ""
if self.isAndroid:
    curLeft = eachEditTextLocation[0]
    curTop = eachEditTextLocation[1]
    curRight = curLeft + eachEditTextLocation[2]
    curBottom = curTop + eachEditTextLocation[3]
    curBounds = "[%s,%s][%s,%s]" % (curLeft, curTop, curRight, curBottom) # [191,2060][430
,2135], '[939,423][1621,558]', '[939,569][1621,701]'
    # foundElement = self.xpathFindElement(curClass=EditTextClass, curBounds=curBounds)
    editTextLocator = {
        "type": "class+bounds",
        "class": EditTextClass, # 'android.widget.EditText'
        "bounds": curBounds, # '[939,423][1621,558]'
    }
    foundElement = self.find_element_Android(editTextLocator)
    # Note: Not use follow wait_element_setText, for it only support type=text
    # isSetOk = self.wait_element_setText(editTextLocator, curInputValue)
    if foundElement:
        foundElement.set_text(curInputValue)

```

## 操作元素

找到元素后，往往会涉及到操作元素，其中常见的一些操作有：

- 点击元素
- (给元素) 输入内容

下面详细介绍如何操作。

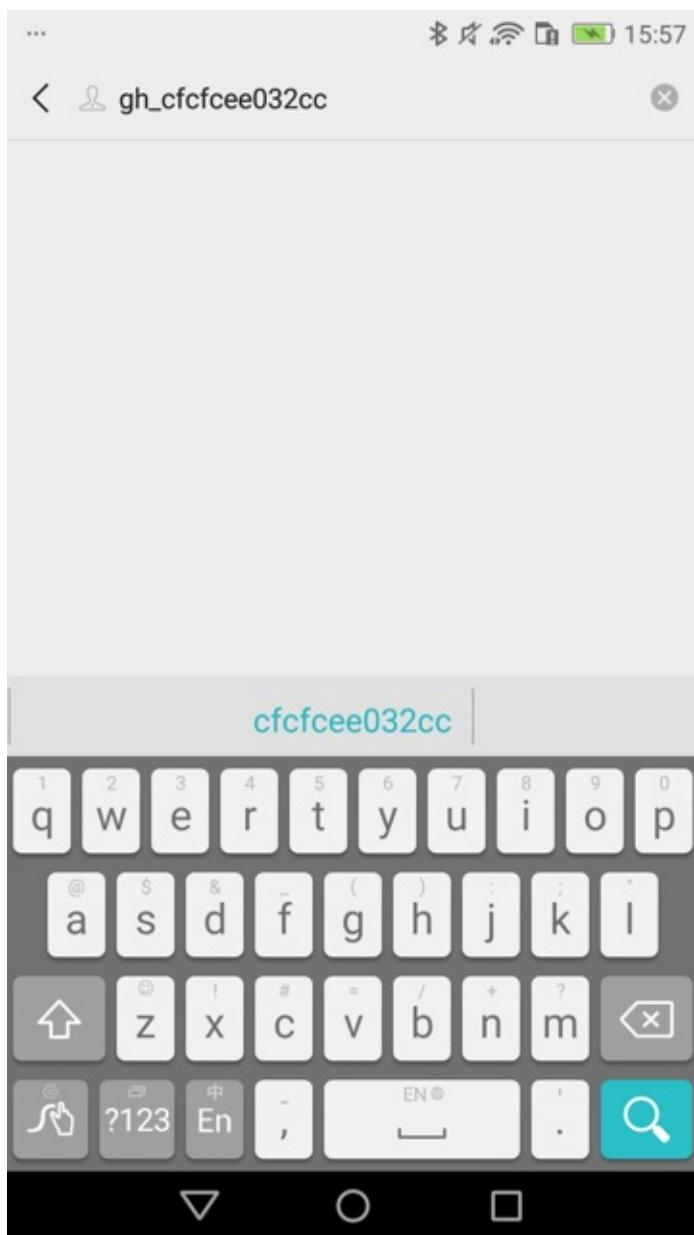
crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2020-08-09 10:14:28

## 点击元素

找到元素后，往往涉及到点击元素。

### 举例：点击搜索按钮

此处，对于安卓手机，华为畅享6S DIG-AL00，当前微信的公众号搜索界面中，已经处于系统自带输入法：华为Skype输入法时



用对应代码：

```
self.driver.send_action("search")
```

可以实现点击对应的 蓝色搜索 按钮，触发搜索，进入搜索结果页面。

详见：

【已解决】uiautomator2中点击华为手机中系统自带Swype的输入法中的搜索按钮

## 点击元素（带等待时间）

```

def wait_element_click_Android(self, locator, wait=0.1):
    foundAndClicked = False

    if isinstance(locator, list):
        self.tap(locator)
        foundAndClicked = True
    elif isinstance(locator, dict):
        locatorType = locator.get("type")
        locatorText = locator.get("text")
        locatorClass = locator.get("class")
        locatorDesc = locator.get("desc")
        locatorId = locator.get("id")
        locatorInstance = locator.get("instance")
        locatorBounds = locator.get("bounds")

        foundElement = None
        hasClicked = False
        if locatorType:
            if locatorType == "text":
                foundElement = self.driver(text=locatorText)
            elif locatorType == "desc":
                foundElement = self.driver(description=locatorDesc)
            elif locatorType == "id":
                foundElement = self.driver(resourceId=locatorId)
            elif locatorType == "id+bounds":
                foundElement = self.xpathFindElement(curClass=locatorClass, curId=locatorId,
, curBounds=locatorBounds)
            elif locatorType == "id+text":
                foundElement = self.driver(resourceId=locatorId, text=locatorText)
            elif locatorType == "id+desc":
                foundElement = self.driver(resourceId=locatorId, description=locatorDesc)
            elif locatorType == "class+instance":
                # foundElement = self.driver(className=locator["class"], instance=locatorI
nstance)
                instanceInt = int(locatorInstance)
                # foundElement = self.driver(className=locatorClass, instance=instanceInt)
                foundElementList = self.driver(className=locatorClass)
                if foundElementList:
                    curIdx = instanceInt
                    shouldMaxNumber = curIdx + 1
                    if foundElementList.count >= shouldMaxNumber:
                        foundElement = foundElementList[curIdx]

                if not foundElement:

```

```

        if locatorText:
            foundElement = self.driver(className locatorClass, text locatorText)
    )

    if not foundElement:
        if locatorBounds:
            # # method 1: click center point
            # centerPoint = self.boundsToCenterPoint(locatorBounds)
            # self.tap(centerPoint)
            # hasClicked = True

            # method 2: find by xpath with bounds
            foundElement = self.xpathFindElement(curClass locatorClass, curId=
locatorId, curBounds locatorBounds)
        elif locatorType == "point":
            centerPoint = self.boundsToCenterPoint(locatorBounds)
            self.tap(centerPoint)
            hasClicked = True
            foundAndClicked = True
    else:
        if locatorText:
            foundElement = self.driver(text locatorText)
        elif locatorDesc:
            foundElement = self.driver(description locatorDesc)
        elif locatorId is not None:
            foundElement = self.driver(resourceId locatorId)
        elif locatorClass and locatorInstance:
            foundElement = self.driver(className locatorClass, instance locatorInstance)

        elif locator.get('wixin_text_matched') is not None:
            # TODO 优化微信公众号异常
            foundElement = self.driver(textContains locator["wixin_text_matched"])

    logging.debug("hasClicked=%s, foundElement=%s", hasClicked, foundElement)
    if not hasClicked:
        isFound = False
        isClickable = None
        hasTimeoutPara = False

        if isinstance(foundElement, u2.xpath.XMLElement):
            hasTimeoutPara = False
            if foundElement:
                isFound = True
                isClickableStr = foundElement.attrib["clickable"]
                isClickableStr = isClickableStr.lower()
                if isClickableStr == "true":
                    isClickable = True
                elif isClickableStr == "false":
                    isClickable = False
            # elif isinstance(foundElement, u2.session.UiObject):
            elif isinstance(foundElement, u2.UiObject):
                hasTimeoutPara = True
                if foundElement and foundElement.exists:

```

```

        isFound = True
        if (foundElement.count > 1):
            foundElement = foundElement[0]

        isClickable = foundElement.info["clickable"]

    if isFound:
        # # for debug: click clickable=false element still working for old android
6.0
        # isClickable = True

        if isClickable:
            if hasTimeoutPara:
                foundElement.click(timeout wait)
            else:
                foundElement.click()
            foundAndClicked = True
        else:
            logging.info("Try auto find and click clickable=true parent for locator=%s", locator)
            logging.debug("clickable=false element info: %s", foundElement.info)

            # curClassName = foundElement.info["className"]
            # curNodeXPath = None
            # if locatorText:
            #     # most case: android.widget.TextView
            #     # sometime: android.view.View
            #     curNodeXPath = "//%s[@text='%s']" % (curClassName, locatorText)
            # elif locatorId:
            #     # {'bounds': '[41,323][1039,727]', 'class': 'android.widget.ImageView', 'id': 'com.ss.android.auto:id/j', 'instance': '10', 'type': 'id'}
            #     curNodeXPath = "//%s[@resource-id='%s']" % (curClassName, locatorId)

            curClassName = None
            curResId = None
            curBoundsStr = None

            # curAttrib = foundElement.attrib
            # AttributeError: 'UiObject' object has no attribute 'attrib'
            if hasattr(foundElement, "attrib"):
                curAttrib = foundElement.attrib
                # {'index': '0', 'text': '', 'resource-id': 'com.netease.newsreader.activity:id/hs', 'package': 'com.netease.newsreader.activity', 'content-desc': '', 'checkable': 'false', 'checked': 'false', 'clickable': 'false', 'enabled': 'true', 'focusable': 'false', 'focused': 'false', 'scrollable': 'false', 'long-clickable': 'false', 'password': 'false', 'selected': 'true', 'visible-to-user': 'true', 'bounds': '[75,2098][141,2134]'}
                curResId = curAttrib["resource-id"]
                curBoundsStr = curAttrib["bounds"]
            else:
                # # for debug
                # self.debugPrintElement(foundElement, "no attrib")
                logging.debug("")

```

```

        curInfo = foundElement.info
        # {'bounds': {'bottom': 2134, 'left': 75, 'right': 141, 'top': 2098},
        'className': 'android.widget.ImageView', 'contentDescription': '',
        'enabled': 'true', 'foc
usable': 'false', 'focused': 'false', 'longClickable': 'false',
        'packageName': 'com.netease.newsreader....activity', 'scrollable': 'false',
        'selected': 'true', 'text': ''}

        if not curClassname:
            curClassname = curInfo["className"] # 'android.widget.ImageView'

        if not curBoundsStr:
            boundsDict = curInfo["bounds"]
            x0 = boundsDict["left"]
            y0 = boundsDict["top"]
            x1 = boundsDict["right"]
            y1 = boundsDict["bottom"]
            curBoundsStr = "[%d,%d][%d,%d]" % (x0, y0, x1, y1)
            # '[75,2098][141,2134]'

        if not curResId:
            if "resourceName" in curInfo:
                curResId = curInfo["resourceName"] # 'com.netease.newsreader.a
ctivity:id/bn5'

        curNodeXPath = self.generateElementXPath(
            curClass curClassname,
            curId=curResId,
            curBounds curBoundsStr,
        )

        if curNodeXPath:
            # # for debug
            # if curResId:
            #     if re.search("(id/a_9)|(id/bcp)", curResId):
            #         self.debugPrintElement(foundElement, "debug find parent
3->5?")
            # else:
            #     self.debugPrintElement(foundElement, "empty resource id")

            foundAndClicked = self.findAndClickNode(curNodeXPath)
        else:
            # TODO: add other type later
            logging.warning("Not click for click will not work for clickable=
else of element: %s, found by locator=%s",
                            foundElement.info, locator)
        else:
            logging.warning("Not click for not found element by locator=%s", locator)

        time.sleep(wait)
        return foundAndClicked

```

调用：

```
if self.isAndroid:  
    foundAndClicked = self.wait_element_click_Android(locator_locator, wait wait)
```

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2021-07-18 09:59:23

## 输入内容

找到元素后，也会遇到需要输入内容的情况。

典型用法是：

```
# 方式1：xpath的set_text方式
searchElementSelector = self.driver.xpath(locatorText)
searchElementSelector.set_text(text)
```

即可输入文字。

后记：已整理成独立函数：

```
def selectorSetText(self, curXPathSelector, inputText):
    # Special: add click to try workaround for 360 pwd EditText input but input to 360 account EditText
    # curXPathSelector.click()
    # curXPathSelector.clear_text()
    selectorSetTextResp = curXPathSelector.set_text(inputText)
    logging.debug("selectorSetTextResp=%s", selectorSetTextResp) # selectorSetTextResp=None

    # 在set_text后，输入法会变成FastInputIME输入法
    # 用下面代码可以实现：关掉FastInputIME输入法，切换回系统默认输入法
    self.driver.set_fastinput_ime(False)
```

调用举例：

```
Qihoo360_Account = "yourAccount"
accountXPath = """//android.widget.EditText[@resource-id="com.qihoo.gamecenter.pluginapk:id/lp_account_input_edit" and @index="1"]"""
accountSelector = self.driver.xpath(accountXPath)
self.selectorSetText(accountSelector, Qihoo360_Account)
```

举例：点击和进入百度搜索页面并输入内容，再触发搜索

用代码：

```
#####
# Open baidu homepage
#####

# trigger into input page

# Method 1
inputUiObj = d(resourceId="com.android.browser:id/b4h", className="android.widget.TextView")
```

```
)  
inputUiObj.click()  
  
## Method 2  
# inputXpathSelector = d.xpath("//android.widget.TextView[@resource-id='com.android.browser:id/b4h'])")  
# inputXpathSelector.click()  
  
# input baidu homr url  
BaiduHomeUrl = "https://www.baidu.com/"  
searchUiObj = d(resourceId: "com.android.browser:id/bq3", className: "android.widget.EditText")  
searchUiObj.set_text(BaiduHomeUrl)  
  
# trigger jump to baidu home  
d.press("enter")
```

实现了：

在浏览器首页中，点击顶部搜索框，进入搜索地址栏输入框页面

输入了百度首页地址后

输入回车键，触发跳转

进入了百度首页

效果：

ATX WEditor    Android    orga4pmzee4ts47t    Connect    Dump Hierarchy    静态  实时

Selected Element  Clear Canvas

Tap Widget(Beta)    Tap    Send Keys

Prop	Value
activity	.BrowserActivity
package	com.android.browser
className	android.widget.EditText
description	
text	百度一下
resourceId	com.android.browser:id/bq3
clickable	true
label	
x, y	160, 83
width	659
height	112
enabled	true
visible	
XPath	//*[@resource-id="com.android.browser:id/bq3"]
position %	(0.323, 0.06) 点击
代码	<input type="checkbox"/> 自动复制代码 <input type="checkbox"/> 强制使用XPath
d(resourceId="com.android.browser:id/bq3")	

POWER    Home    Back    Menu

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook 最后更新: 2021-07-18 09:59:23

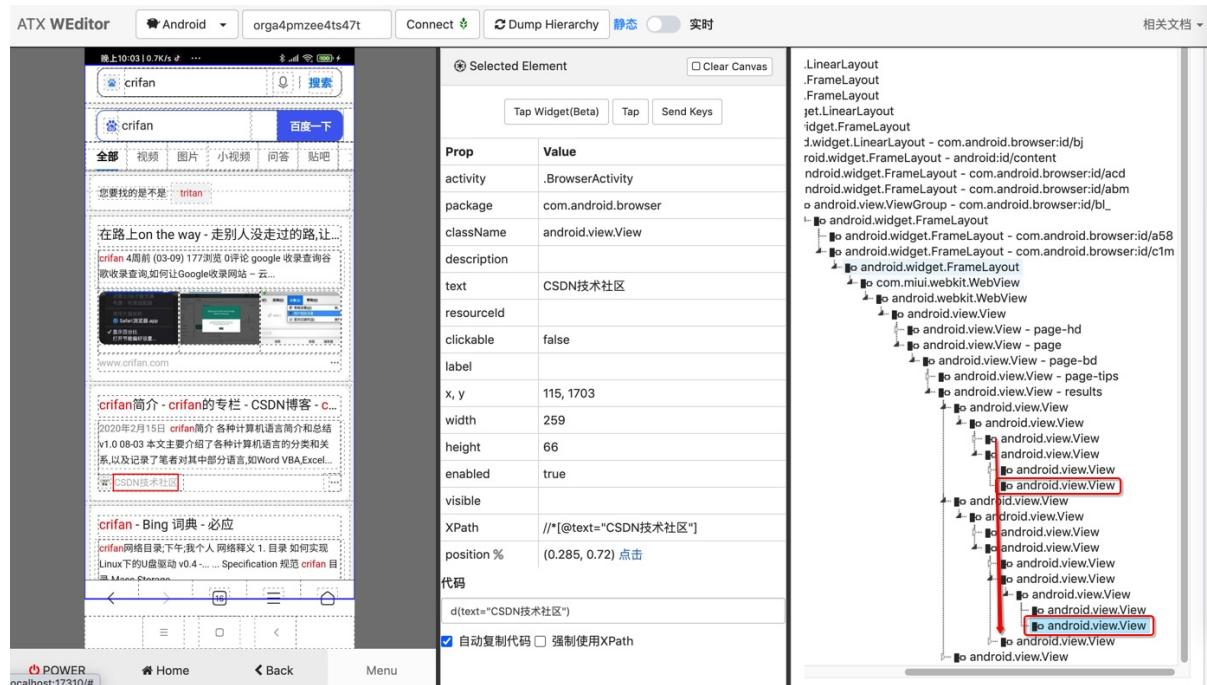
# 获取元素属性

获取元素属性，一般可以用：

- `someElement.attrib`：返回一个 `dict`
- `someElement.text`：获取元素文本值

## 举例：获取百度搜索结果页面的元素列表中的元素属性

对于（调试）页面：



用代码：

```
# Xpath chain search can find elements
titleButtonElementList = d.xpath("//*[@resource-id='results']//android.widget.Button[@clickable='true']").all()
titleButtonNum = len(titleButtonElementList)
print("Found %s search result title" % titleButtonNum)

# descriptionElementList = d.xpath("//*[@resource-id='results']/android.view.View[1]/android.view.View[1]/android.view.View[2]/android.view.View[1]/android.view.View[1]/android.view.View[1]/android.view.View[1]/android.view.View[1]").all()
descriptionElementList = d.xpath("//*[@resource-id='results']/android.view.View/android.view.View[1]/android.view.View[2]/android.view.View[1]/android.view.View[1]/android.view.View[1]/android.view.View[1]/android.view.View[1]").all()
descriptionNum = len(descriptionElementList)
print("Found %s description" % descriptionNum)
```

```

# # sourceWebsiteElementList = d.xpath('//*[@resource-id="results"]/android.view.View/android.view.View[1]/android.view.View[2]/android.view.View[1]').all()
# sourceWebsiteElementList = d.xpath('//*[@resource-id="results"]/android.view.View/android.view.View[1]/android.view.View[2]/android.view.View[2)').all()
# sourceWebsiteNum = len(sourceWebsiteElementList)
# print("Found %s source website" % sourceWebsiteNum)

for curIdx, eachTitleButtonElement in enumerate(titleButtonElementList):
    curNum = curIdx + 1
    print("%s [%d/%d] %s" % ("-"*20, curNum, titleButtonNum, "-"*20))
    # eachTitleButtonElemAttrib = eachTitleButtonElement.attrib
    # print("title attrib: %s" % eachTitleButtonElemAttrib)
    # curTitle = eachTitleButtonElemAttrib["text"]
    curTitle = eachTitleButtonElement.text
    print("title=%s" % curTitle)

    curDescriptionElem = descriptionElementList[curIdx]
    curDescription = curDescriptionElem.text
    print("description=%s" % curDescription)

    # curSourceWebsiteElem = sourceWebsiteElementList[curIdx]
    # curSourceWebsite = curSourceWebsiteElem.text
    # print("curSourceWebsite=%s" % curSourceWebsite)

```

可以搜索到 title 外，还能输出 描述

输出结果：

```

Found 3 search result title
Found 3 description
----- [1/3] -----
title 在路上on the way - 走别人没走过的路,让别人有路可走
description crifan 4周前 (03-09) 177浏览 0评论 google 收录查询谷歌收录查询,如何让Google收录网站
- 云...
----- [2/3] -----
title crifan简介 - crifan的专栏 - CSDN博客 - crifan
description 2020年2月15日crifan简介 各种计算机语言简介和总结 v1.0 08-03 本文主要介绍了各种计算机
语言的分类和关系,以及记录了笔者对其中部分语言,如Word VBA,Excel...
----- [3/3] -----
title crifan - Bing 词典 - 必应
description crifan网络目录;下午;我个人 网络释义 1. 目录 如何实现Linux下的U盘驱动 v0.4 - ... ...
Specification 规范 crifan 目录 Mass Storage ...

```

效果：

```

uiautomator2DemoBaiduSearch.py — BaiduSearchAutomation
RUN AND DEBUG  Python: 当前文件  ⌂ ...  uiautomator2DemoBaiduSearch.py ×  baidu_search_crifan_result_pageSource_reloaded.xml
VARIABLES
Locals
special variables
BaiduHomePage: 'https://www.baidu.com'
Browser_XiaomiBuiltIn: 'com.android.browser'
EnterKey: 'enter'
MustShowTabName: '我的'
baiduSearchButtonUiObj: <uiautomator2.session._.
baiduSearchKeywordUiObj: <uiautomator2.session._.
browserPackage: 'com.android.browser'
curDescription: 'crifan网络目录;下午;我个人 网络释义'
curDescriptionElem: <uiautomator2.xpath.XMLEle...
curIdx: 2
curNum: 3
curTitle: 'crifan - Bing 搜索 - 必应'
d: uiautomator2 object for 127.0.0.1:58165
descriptionElementList: [<uiautomator2.xpath._.
descriptionNum: 3
eachTitleButtonElement: <uiautomator2.xpath.XM...
inputUIObj: <uiautomator2.session.UiObject obj...
searchStr: 'crifan'
searchUiObj: <uiautomator2.session.UiObject obj...
time: <module 'time' (built-in)
titleButtonElementList: [<uiautomator2.xpath._.
titleButtonNum: 3
u2: <module 'uiautomator2' from '/Users/limao/...
Globals

WATCH
CALL STACK  'XPATH' OBJECT HAS NO ATTRIBUTE 'PARENT'
BREAKPOINTS
■ Raised Exceptions
☒ Uncaught Exceptions
uiautomator2DemoBaiduSearch.py  uiauto...  182
Python 3.8.0 64-bit  0 △ 0  Python: 当前文件 (BaiduSearchAutomation)

```

```

TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE  2: Python Debug Console  +  ⌂  ^  ×
Found 3 search result title
Found 3 description
[1/3] -----
title=在路上 on the way - 走别人没走过的路,给别人有路可走
description=crifan 4周前 (03-09) 177浏览 0评论 google 收录查询谷歌收录查询,如何让Google收录网站 - 云...
[2/3]
title=crifan简介 - crifan的专栏 - CSDN博客 - crifan
description=2020年2月15日crifan简介 各种计算机语言简介和总结 v1.0 08-03 本文主要介绍了各种计算机语言的分类和关系,以及记录了笔者对其中部分语言,如Word VBA,Excel...
[3/3]
title=crifan - Bing 搜索 - 必应
description=crifan网络目录;下午;我个人 网络释义 1. 目录 如何实现Linux下的U盘驱动 v0.4 -... ... Specification 规范 crifan 目录 Mass Storage
e ...
Ln 163, Col 61  Spaces: 4  UTF-8  LF  Python  ⌂  □

```

说明：

由于来源网站source website的值，不规则，不好定位，放弃获取。

此处是即使获取到，也是只有2个（页面是就显示了2个）

且还都是空（没有获取到真正要的值）。此处暂时不深究了。有机会再深究。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2021-07-18 09:59:23

## 按键

操作安卓手机，常会遇到，模拟按键输入。

可以用 `press`，比如：

```
TriggerSearchKey = "enter" # work
d.press(TriggerSearchKey)
```

## 举例

代码：

```
#####
# Input text
#####
searchStr = "crifan"

baiduSearchKeywordUiObj = d(resourceId "index-kw", className "android.widget.EditText")
baiduSearchKeywordUiObj.set_text(searchStr)
print("Inputed baidu search text %s" % searchStr)

#####
# Trigger baidu search
#####

# # Method 1: press key
TriggerSearchKey = "enter" # work
# # TriggerSearchKey = "search" # not work
# # TriggerSearchKey = "go" # not work
# # TriggerSearchKey = "done" # not work
d.press(TriggerSearchKey)
print("Emulated press key %s" % TriggerSearchKey)

# Method 2: find 百度一下 button then click
# baiduSearchButtonUiObj = d(resourceId="index-bn", className="android.widget.Button")
# baiduSearchButtonUiObj.click()
print("Clicked baidu search button")
```

可以实现：

给百度首页中

输入要搜索的文字

并（模拟按下回车键，或，找到百度一下按钮并点击）触发搜索

效果：

localhost:17310

ATX WEditor    Android    orga4pmzee4ts47    Connect    Dump Hierarchy    静态    实时

Prop	Value
activity	.BrowserActivity
package	com.android.browser
className	android.widget.EditText
description	
text	crifan
resourceId	kw
clickable	true
label	
x, y	132, 248
width	536
height	127
enabled	true
visible	
XPath	<code>//*[@resource-id='kw']</code>
position %	(0.355, 0.136) 点击

代码

```
d(resourceId="kw")
```

自动复制代码  强制使用XPath

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook 最后更新: 2021-07-18 09:59:23

## 屏幕

此处整理和屏幕相关操作

### 点击 (屏幕 坐标)

```
self.driver.click(x, y)
```

### 长按

```
self.driver.long_click(x, y, duration 1.5)
```

### 滑动

```
# 等待时间
SwipeDuration_Android = 0.3

SwipeDirectionBounds = [338, 333, 38, 333]
curSession.swipe(SwipeDirectionBounds[0], SwipeDirectionBounds[1], SwipeDirectionBounds[2],
SwipeDirectionBounds[3], SwipeDuration_Android)

self.driver.swipe(SwipeDirectionBounds[0], SwipeDirectionBounds[1], SwipeDirectionBounds[2]
, SwipeDirectionBounds[3], SwipeDuration_Android)
```

(从当前屏幕) 返回上一页

```
self.driver.press("back")
```

### (屏幕) 坐标值

boundsToCenterPoint: 从bounds算出中间坐标值

```
def boundsToCenterPoint(self, boundsStr):
    """
    从bounds转换出中间点位置坐标

    Example:
        bounds: '[156,1522][912,2027]'
        return: [534, 1774]
```

```

"""
filterStr = re.sub('\[|]', "", boundsStr)
boundStrList = filterStr.split()
boundMap = map(int, boundStrList)
boundIntList = list(boundMap)
x0 = boundIntList[0]
y0 = boundIntList[1]
x1 = boundIntList[2]
y1 = boundIntList[3]
centerPoint = [(x1 + x0)//2, (y1 + y0)//2]
return centerPoint

```

调用：

```

centerPoint = self.boundsToCenterPoint(locatorBounds)
self.tap(centerPoint)

```

## 当前屏幕

针对于当前屏幕，最常见的几个动作是：

- 截图=截屏
- 获取(当前)页面源码(xml)

### 给当前屏幕截图

核心代码：

```
fullImgFilePath = self.driver.screenshot(fullImgFilePath)
```

举例：

```

fullImgFilePath = 'debug/GameScreenshot/20191209_171115.png'
fullImgFilePath = self.driver.screenshot(fullImgFilePath)

```

`getCurPageSource`: 获取当前屏幕画面对应的xml源码

函数：

```

def getCurPageSource(self):
    # curPageSrcXml = self.driver.dump_hierarchy()
    curPageSrcXml = self.driver.dump_hierarchy(compressed=False, pretty=False)

    # output, exitCode = self.driver.shell(["adb", "shell", "uiautomator", "dump"])
    # output, exitCode = self.driver.shell(["uiautomator", "dump"])
    # output, exitCode = self.driver.shell("uiautomator dump")
    # output, exitCode = self.driver.shell(["shell", "uiautomator", "dump"])

```

```
# curPageSrcXml = output

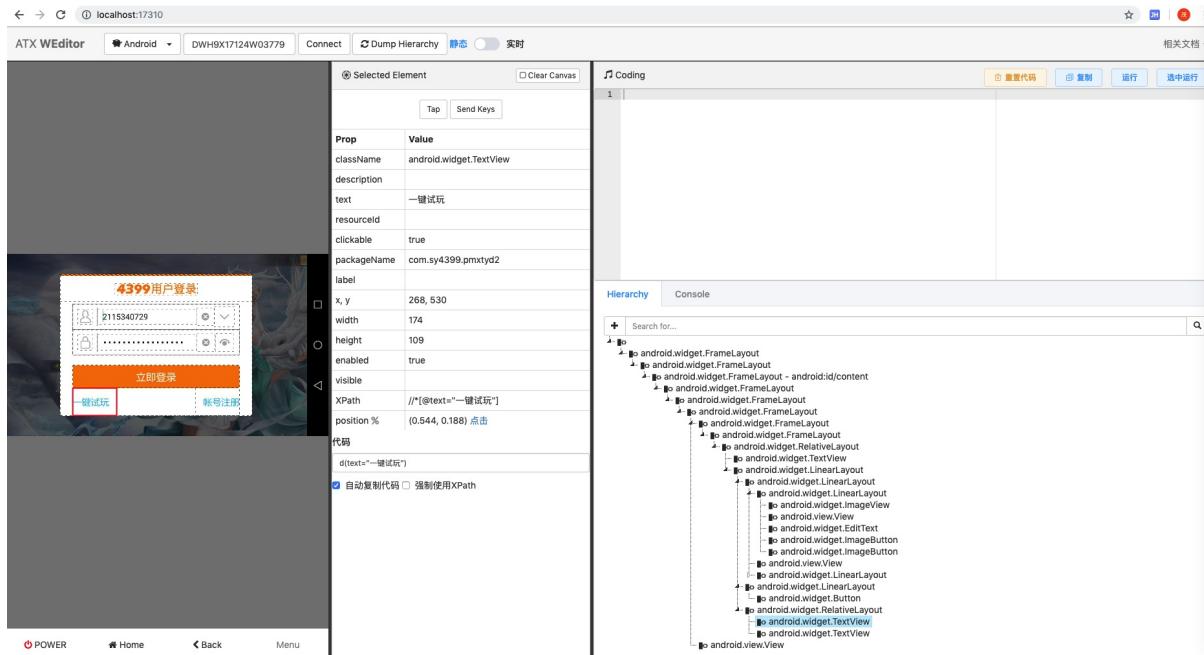
return curPageSrcXml
```

调用：

```
curPageSrcXml = self.getCurPageSource()
```

举例：

对于下图中左边的登录界面：



用：

```
page_source = self.driver.dump_hierarchy(compressed=False, pretty=False)
```

导出的源码是：

```
<?xml version='1.0' encoding='UTF-8' standalone='yes'?>
<hierarchy rotation="1">
    <node index="0" text="" resource-id="" class="android.widget.FrameLayout" package="com.sy4399.pmxtyd2" content-desc="" checkable="false" checked="false" clickable="false" enabled="true" focusable="false" focused="false" scrollable="false" long-clickable="false" password="false" selected="false" bounds="[0,0][1196,720]">
        ...
        <node NAF="true" index="4" text="" resource-id="" class="android.widget.ToggleButton" package="com.sy4399.pmxtyd2" content-desc="" checkable="true" checked="false" clickable="true" enabled="true" focusable="true" focused="false" scrollable="false" long-clickable="false" password="false" selected="false" bounds="[834,314][906,386]" />
    </node>
```

```
        </node>
        <node index="1" text="" resource-id="" class=""
    android.widget.LinearLayout" package="com.sy4399.pmxtyd2" content-desc="" checkable="false"
    checked="false" clickable="false" enabled="true" focusable="false" focused="false" scroll
    able="false" long-clickable="false" password="false" selected="false" bounds="[268,440][92
    8,530]">
            <node index="0" text="立即登录" resource-id=
    "" class="android.widget.Button" package="com.sy4399.pmxtyd2" content-desc="" checkable="f
    alse" checked="false" clickable="true" enabled="true" focusable="true" focused="false" scr
    ollable="false" long-clickable="true" password="false" selected="false" bounds="[268,440][
    928,530]" />
        </node>
    ...
        </node>
    </node>
    </node>
    </node>
    <node index="1" text="" resource-id="" class="android.view.View"
    package="com.sy4399.pmxtyd2" content-desc="" checkable="false" checked="false" clickable=
    "false" enabled="true" focusable="false" focused="false" scrollable="false" long-clickable=
    "false" password="false" selected="false" bounds="[218,80][978,86]" />
        </node>
    </node>
    </node>
    </node>
</node>
</hierarchy>
```



crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2021-07-18 09:59:23

# APP

## 安装app

```

def install_app_Android(self, item, packages=None):
    appAccount = item[0]
    appPackage = item[1]
    appMainActivity = item[2]
    apkFilePath = item[3]

    if packages is None:
        packages = self.get_packages()
    if appPackage in packages:
        logging.info("AppName {0} is already installed".format(appAccount))
    else:
        logging.info("start to install app in {}".format(os.path.basename(self.arg_options.task)))
    # show current adb command version, makesure is latest version
    # os.system("adb --version")
    # Android Debug Bridge version 1.0.41
    # Version 30.0.5-6877874
    # Installed as /Users/limao/dev/tools/android/adb/platform-tools/adb

    # isUseShortCmd = True # use short command to try to avoid install stuck
    isUseShortCmd = False # use default long abd install command

    if isUseShortCmd:
        # sometime will stuck, reason: maybe command length too large, > 255?
        # so copy to temp folder -> to reduce command string length
        tmpFolder = tempfile.TemporaryDirectory()
        logging.info("tmpFolder=%s", tmpFolder)
        # tmpFolder=<TemporaryDirectory '/var/folders/gt/5868sbcd1jq4rxvryqhy2_1sz8n0s3/T/tmp_yk15rlz'>
        tmpFolderName = tmpFolder.name
        logging.info("tmpFolderName=%s", tmpFolderName)
        # tmpFolderName=/var/folders/gt/5868sbcd1jq4rxvryqhy2_1sz8n0s3/T/tmp_yk15rlz
        apkFileName = os.path.basename(apkFilePath)
        logging.info("apkFileName=%s", apkFileName)
        # apkFileName=20201202_fengyun_0192LeiMoChuanShuo_gameApp_Android_0.apk
        tmpApkFile = os.path.join(tmpFolderName, apkFileName)
        logging.info("tmpApkFile=%s", tmpApkFile)
        # tmpApkFile=/var/folders/gt/5868sbcd1jq4rxvryqhy2_1sz8n0s3/T/tmp_yk15rlz/20201202_fengyun_0192LeiMoChuanShuo_gameApp_Android_0.apk
        logging.info("Copy %s to tmp file %s", apkFilePath, tmpApkFile)
        # Copy /Users/limao/dev/xxx/crawler/appAutoCrawler/AppCrawler/task/20201202_fengyun_0192LeiMoChuanShuo/20201202_fengyun_0192LeiMoChuanShuo_gameApp_Android_0.apk to tmp file /var/folders/gt/5868sbcd1jq4rxvryqhy2_1sz8n0s3/T/tmp_yk15rlz/20201202_fengyun_0192LeiMoChuanShuo_gameApp_Android_0.apk
        copyfile(apkFilePath, tmpApkFile)

```

```

# copy2(apkFilePath, tmpFolderName)

    curApkFile = tmpApkFile
else:
    curApkFile = apkFilePath
    # os.system("adb -s {0} install {1}".format(self.device, apkFilePath))

# show file size
fileSizeInt = os.path.getsize(curApkFile) # 259106541
fileSizeStr = CommonUtils.formatSize(fileSizeInt) # '247.1MB'
logging.info("file size: %s", fileSizeStr) # file size: 247.1MB

# adbInstallCmd = "adb -s {0} install -r {1}".format(self.device, tmpApkFile)
# adbInstallCmd = "adb -s {0} install -r {1}".format(self.device, apkFilePath)
# installPara = ""
# installPara = "-r"
installPara = "-r -f"
adbInstallCmd = "adb -s {0} install {1} {2}".format(self.device, installPara, curApkFile)
# adbInstallCmd = "adb shell pm install -s {0} {1}".format(self.device, curApkFile)
# adbInstallCmd = "adb shell pm install -s {0} -r {1}".format(self.device, curApkFile)
# adbInstallCmd = "adb shell pm install -s {0} -f {1}".format(self.device, curApkFile)
# adb -s hmucae175ptk7szs install -r /var/folders/gt/5868sbcd1jq4rxvryqhy2_1sz8n0s3/T/
tmpq3ypjfgd/20201202_fengyun_0192LeiMoChuanShuo_gameApp_Android_0.apk
# length=243 command:
# adb -s hmucae175ptk7szs install -r /Users/limao/dev/xxx/crawler/appAutoCrawler/AppCr
awler/task/20201202_fengyun_0192LeiMoChuanShuo/20201202_fengyun_0192LeiMoChuanShuo_gameApp
_Android/20201202_fengyun_0192LeiMoChuanShuo_gameApp_Android_0.apk
# 'adb -s hmucae175ptk7szs install -r -f /Users/limao/dev/xxx/crawler/appAutoCrawler/A
ppCrawler/task/20201203_fengyun_0004TiTangMeng/20201203_fengyun_0004TiTangMeng_gameApp_A
ndroid/20201203_fengyun_0004TiTangMeng_gameApp_Android_0.apk'

logging.info("Run length=%d, command: %s", len(adbInstallCmd), adbInstallCmd)
# Run length=153, command: adb -s hmucae175ptk7szs install -r /var/folders/gt/5868sbcd
1jq4rxvryqhy2_1sz8n0s3/T/tmpq3ypjfgd/20201202_fengyun_0192LeiMoChuanShuo_gameApp_Android_0
.apk
os.system(adbInstallCmd)

```

调用：

```

if self.isAndroid:
    return self.install_app_Android(item, packages)

```

## 启动app

对于app：

- 五子棋经典版
  - 包名： com.fingertip.fivechess.mi

- 主页面: `.StartAct`

启动代码:

```
appPackage = "com.fingertip.fivechess.mi"
appActivity = ".StartAct"
self.driver.app_start(appPackage, activity appActivity, stop True)
```

## 卸载app

```
def uninstallApp_Android(self, item):
    # 卸载安装包
    appPackage = item[1]
    adbUninstallCmd = "adb -s %s uninstall %s" % (self.device, appPackage)
    logging.info(adbUninstallCmd)
    os.system(adbUninstallCmd)
    logging.info("Uninstalled Android app %s", appPackage)
```

调用:

```
if self.isAndroid:
    self.uninstallApp_Android(item)
```

## 获取app信息

### 获取当前正在运行的app的包名和activity

```
def get_PackageActivity_Android(self):
    # adb直接获取当前活跃app及activity
    package, activity = "", ""
    cmds = ['dumpsys activity |grep {}'.format(item) for item in ['mFocusedActivity', 'mResumedActivity']]
    for cmd in cmds:
        output = self.driver.shell(cmd).output
        result = re.search("u0(.*)/", output)
        package = result.group(1).strip() if result else ""
        result = re.search("/(.*)\s", output)
        activity = result.group(1).strip() if result else ""
        if package and activity:
            return package, activity
    return package, activity
```

调用:

```
package, activity = get_PackageActivity_Android()
```

## 举例：获取几个浏览器app的信息

可以用 uiautomator2 的 `app_current()`

比如，代码：

```
curApp = d.app_current()  
print("curApp=%s" % curApp)
```

输出：

- 小米 内置浏览器
  - 图
  -



热点 **推荐** 视频 免费小说 小视频 抗 三

潮头观澜 | 新的关头 再探改革

置顶 央视新闻

八闽大地上的奋斗与深情

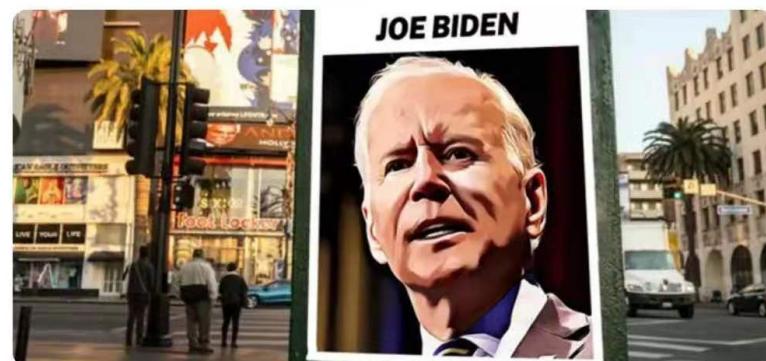
置顶 央视新闻

新时代党的创新理论大众化的又一佳作

人民网

耐克市值蒸发687亿！一个对中国棉花的错误

决定，赔光3个月业绩



军事评论 195条评论



小米11和它谁狠？来，Redmi K40免费“拿”！



小米商城 广告

查看详情



主页



视频



窗口



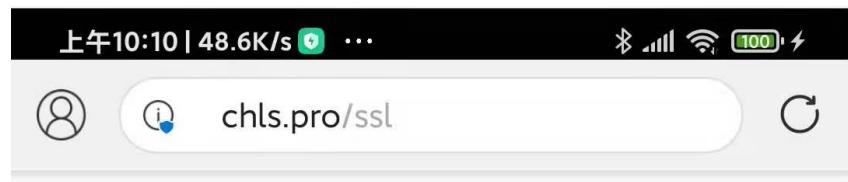
菜单



我的



- 信息
  - { 'package': 'com.android.browser', 'activity': '.BrowserActivity', 'pid': 20194}
- Edge
  - 图
    -



- 信息
  - curApp={'package': 'com.microsoft.emmx', 'activity': 'org.chromium.chrome.browser.ChromeTabbedActivity', 'pid': 21340}
- Firefox
  - 图
    -

上午10:11 | 94.5K/s ...

Firefox Browser

∞

## 欢迎使用 Firefox！

使用您的 Firefox 账户开始同步书签、历史记录等数据。

[详细了解](#)

[登录 Firefox](#)

### 认识 Firefox

全自动的隐私保护

可通过隐私与安全设置，拦截跟踪器、恶意软件和企业在网上对您的跟踪。

标准（默认）  
少拦截一些跟踪器，页面将可正常加载。

严格  
拦截更多跟踪器和弹窗。页面加载速度更快，但是某些功能可能无法正常工作。

### 选择您的主题

启用深色模式，既省电又护眼。

搜索或输入网址

0 :

≡ ☒ <

- 信息
  - curApp={'package': 'org.mozilla.firefox', 'activity': 'org.mozilla.fenix.HomeActivity', 'pid': 21817}
- UC浏览器
  - 图
    -

上午10:11 | 756K/s ... 100%

10
苏州工业园区 多云

搜索或输入网址

小游戏

小说

搞笑段子

携程

斗地主

同城

新浪

搜索

唯品会

苏宁

58生活

酷网址

**温暖**

置顶 新华社

**这些年, 总书记牵挂的民生事**

置顶 人民日报

**“中国是全球贸易复苏的重要贡献者”**

环球网

**重磅! 苏州机场接近获批, 楼市再度开启上涨通道**

小说推荐life 88评

**华为好也不要盲目买, 这4款华为手机公认值得买, 用三五年不过时**

拼多多 广告

头条
  
视频
  
菜单
  
小视频
  
我的

≡
□
<

- 信息
  - curApp={'package': 'com.UCMobile', 'activity': 'com.uc.browser.InnerUCMobile', 'pid': 22173}
- QQ浏览器
  - 图
    -



关注 推荐 视频 热点 免费小说 苏州 N ≡

## 第一观察 | 总书记倡绿植绿，来自太空的观察

置顶 新华社新闻

## “铿锵奋进的脚步 续写更多春天的故事”网络主题宣传

置顶 人民网

## 聆听“国企工匠”的初心故事

光明网

## 中美交锋后，澳加军舰现身南海 解放军一大动作突然炸锅 引爆全球



热点 海纳新闻 398评 15小时前



## “你不对付中国，我就对付你！”美英日加等七国强势放出狠话



首页

免费小说

烟火

文件

我的



- 信息

- curApp={'package': 'com.tencent.mtt', 'activity': '.MainActivity', 'pid': 24304}

## 相关辅助信息

### 获取当前安卓手机中已安装的app的包名等信息

可以通过几种方式：

- 手动点击和查看应用信息
- 用uiautomator2获取当前运行的app的信息
- 用weditor查看当前界面中app某元素的属性中的package值
- 用adb查看

```
adb shell pm list packages
```

获得app的包名信息。

比如：

各种浏览器app的包名：

- 小米的内置浏览器： com.android.browser
- Edge： com.microsoft.emmx
- Firefox： org.mozilla.firefox
- UC浏览器： com.UCMobile
- QQ浏览器： com.tencent.mtt

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2021-07-18 09:59:23

## 其他功能

此处整理其他一些常见的功能。

### 长按

```
self.driver.long_click(x,y,duration 1.5)
```

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2021-07-18 09:59:23

## 相关

uiautomator2开放期间，往往会涉及到一些其他一些内容，此处把相对独立的部分整理出来，单独解释，方便查阅和理解。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2020-08-09 10:14:28

## weditor

折腾u2期间，少不了要调试设备当前的页面，以及希望了解其中的元素和细节。

这时候，同一个作者开发的，用于辅助u2的 `weditor`，就可以派上用场了。

- 主页
  - Github
    - [openatx/weditor: web editor for atx](#)

## 安装

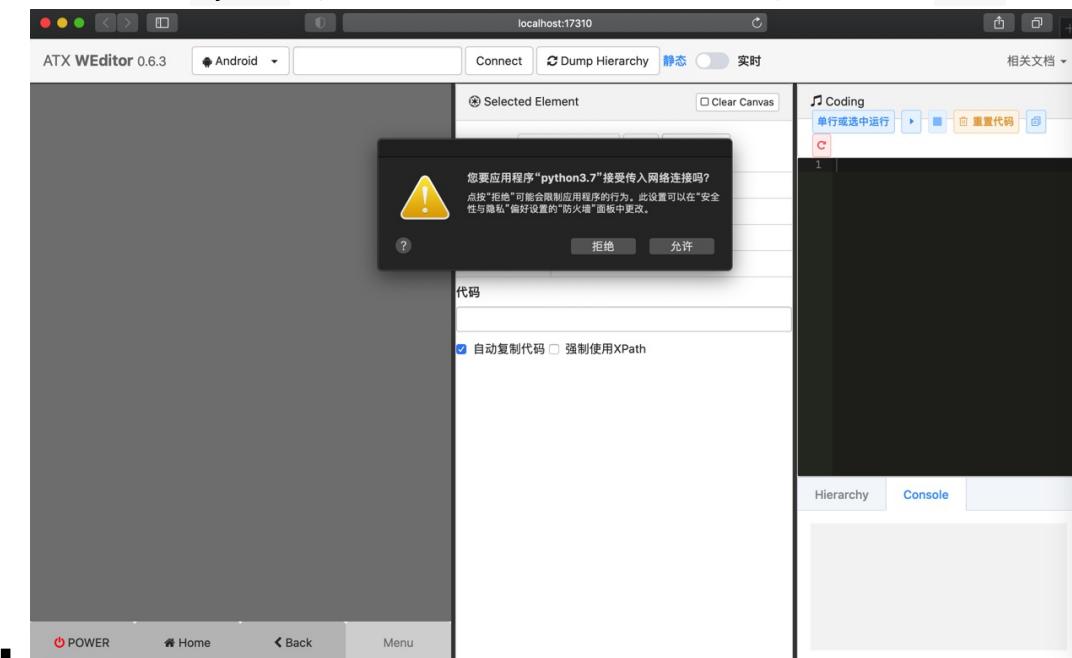
```
pip install -U weditor
```

- 如果是用 `pipenv`，则是: `pipenv install weditor`
- 如果是已安装过，要升级到最新版，则是: `pip install -U weditor`

## 启动

```
weditor
```

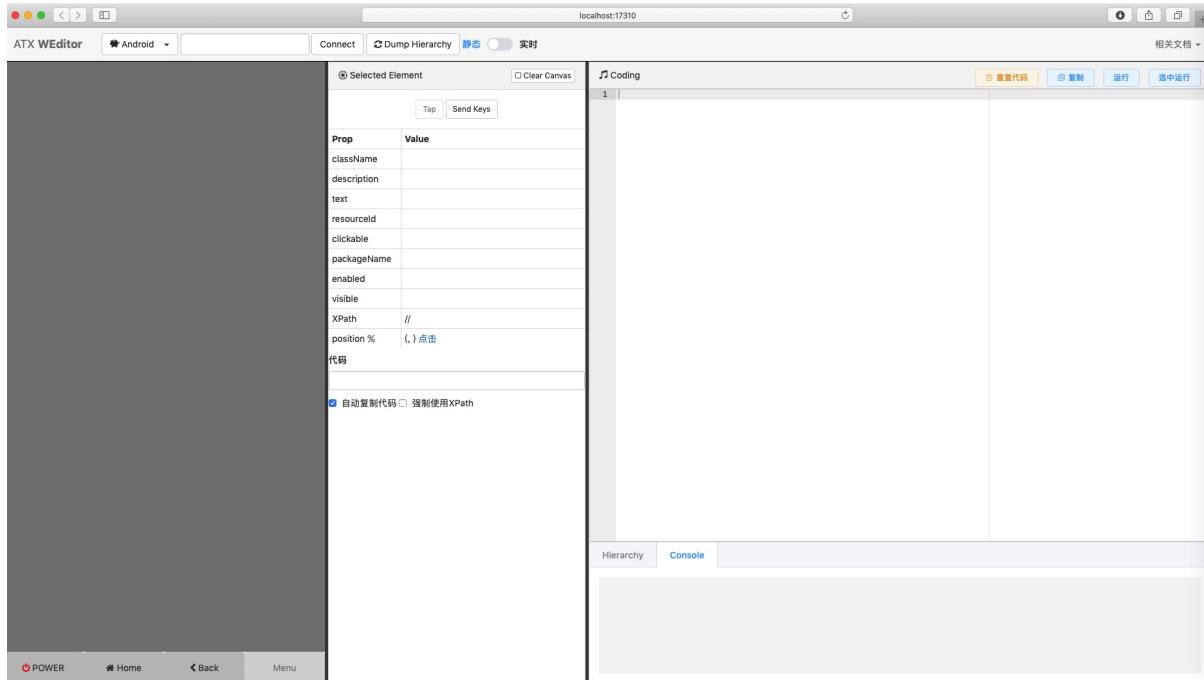
- 注
  - 旧启动方式是: `python -m weditor`
  - Mac中第一次启动 Python 时，可能会遇到申请使用网络的权限提示，注意要点击 允许



会自动调用浏览器并打开网址:

<http://localhost:17310>

效果：



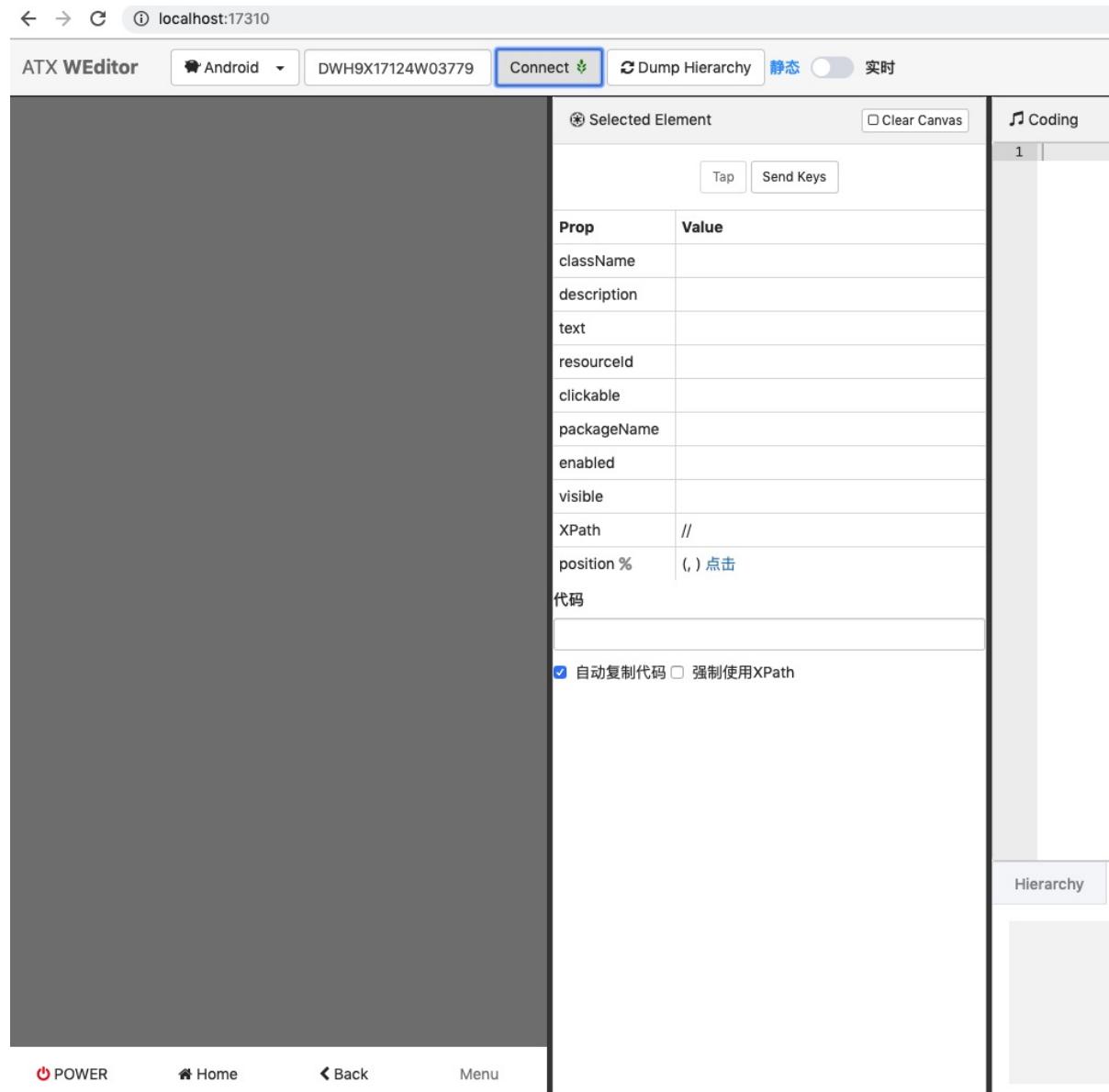
## 使用

概述：

- 连接设备
  - 输入（从adb devices中得到的）安卓设备id
  - 点击Connect，看到绿色 红绿灯标识，标识连接上了。
- 看安卓设备画面
  - 点击 实时，即可看到安卓手机的实时画面。
  - 如果只是想要调试页面元素，则可以切换为 静态
- 查看属性
  - 点击画面中元素，即可看到属性
- 查看元素结构
  - 点击顶部的 Dump Hierarchy，点击右边切换到 Hierarchy，可以看到最新页面的结构

详解：

输入 安卓设备的id 后，点击 Connect 连接设备：



## 用adb devices查看安卓设备的id

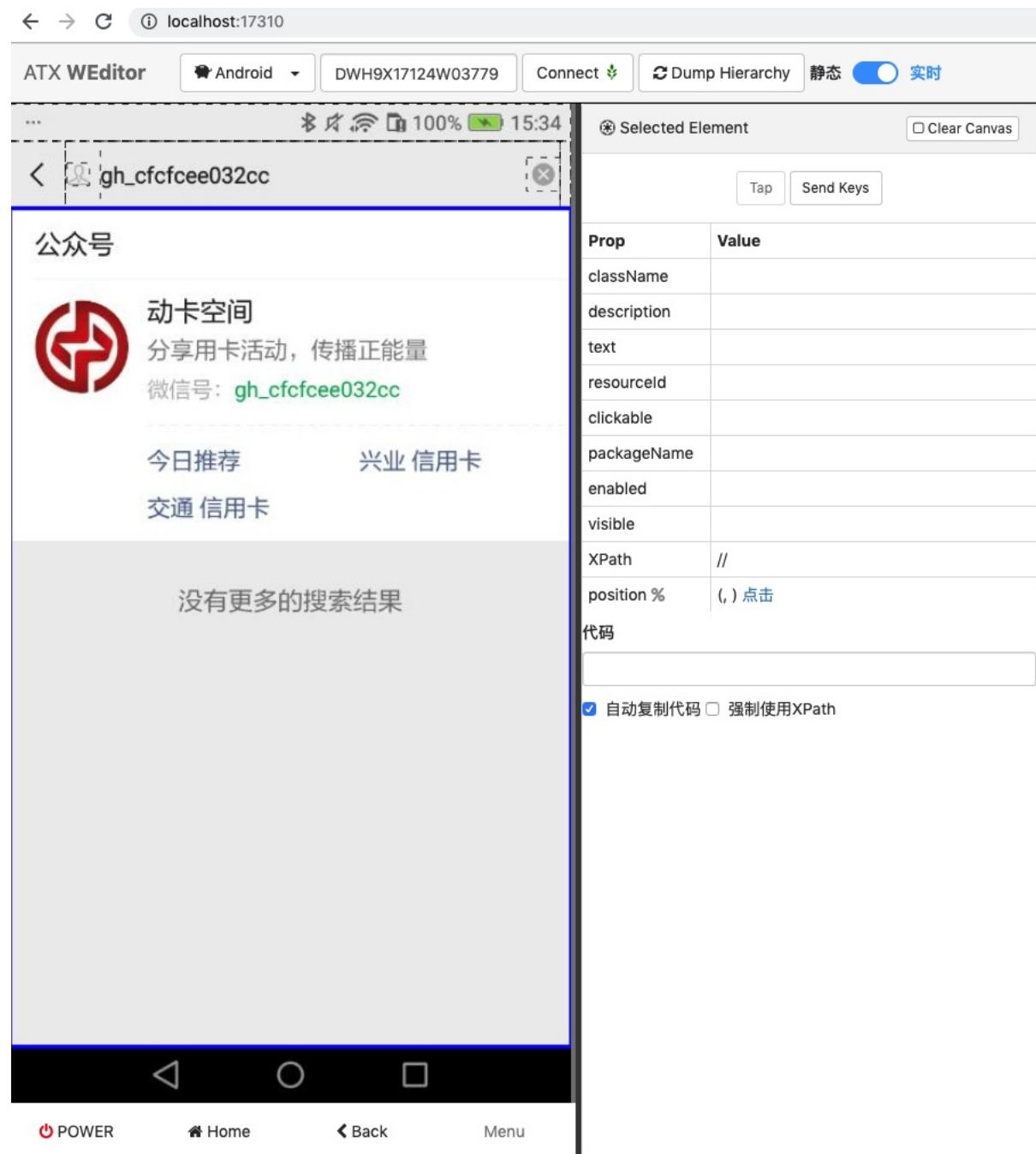
注：关于安卓手机的ID=UDID=序列号，可以用 `adb devices` 获取到

举例：

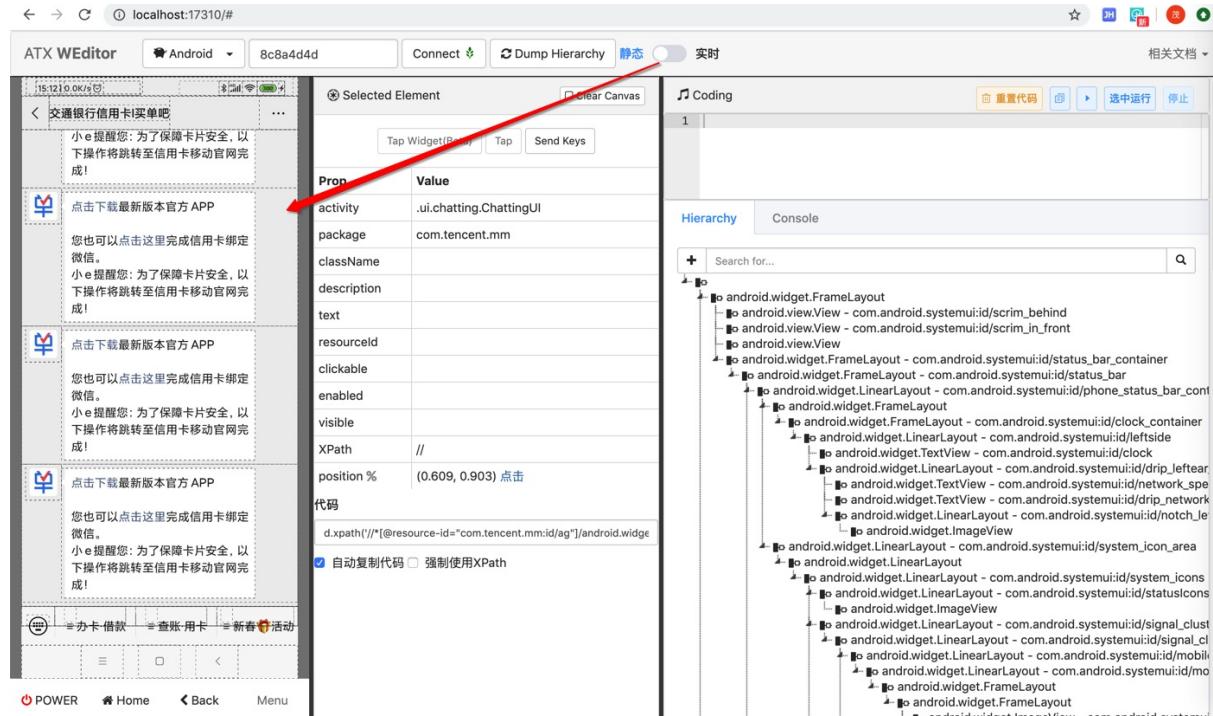
```
~ adb devices
List of devices attached
orga4pmzee4ts47t    device
```

其中的 `orga4pmzee4ts47t` 就是安卓设备的id，也成为 序列号 或 UDID

然后 多次在 静态 实时 直接切换几次，最后一次点击 静态，稍等片刻，就能看到页面内容了：

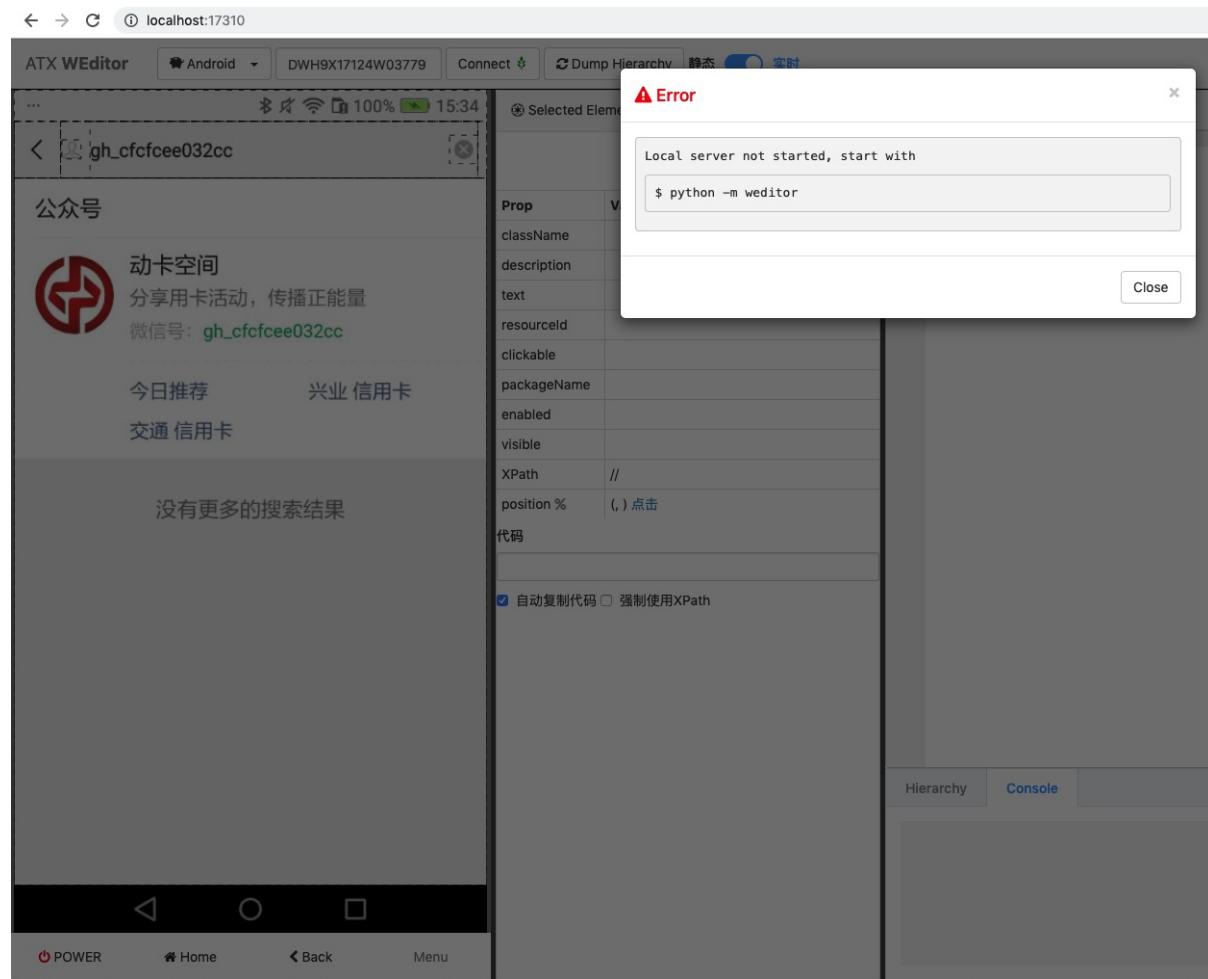


和：



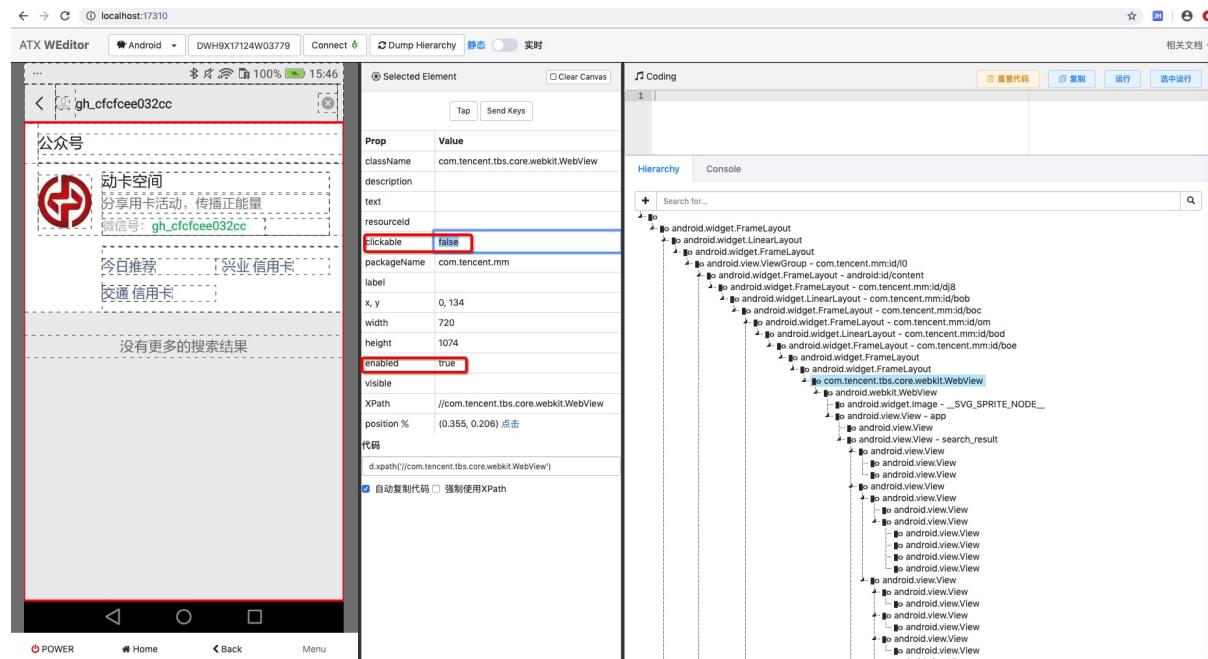
## 报错可忽略

注意，切换期间偶然会报错：



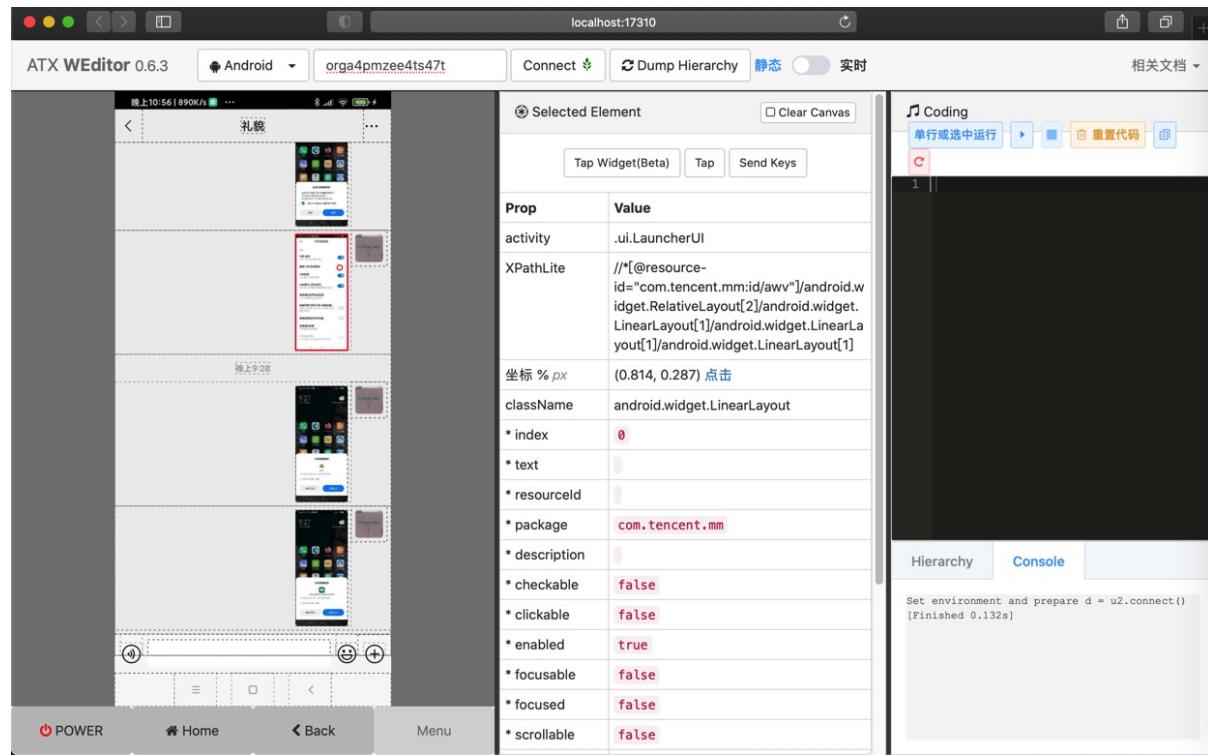
不用理会，关闭弹框，多试几次即可。

点击 Dump Hierarchy 后，能查看到页面的结构：



## 查看元素属性

以及最常用的，点击查看元素属性：



可以拷贝出属性：

```

Selected Element Clear Canvas
Tap Widget(Beta) Tap Send Keys
Prop      Value
activity    .ui.LauncherUI
XPathLite   //*[@resource-id="com.tencent.mm:id/awv"]/android.widget.RelativeLayout[2]/an
droid.widget.LinearLayout[1]/android.widget.LinearLayout[1]/android.widget.LinearLayout[1]
坐标 % px  (0.814, 0.287) 点击
className   android.widget.LinearLayout
* index     0
* text
* resourceId
* package   com.tencent.mm
* description
* checkable  false
* clickable  false
* enabled    true
* focusable  false
* focused    false
* scrollable false
* longClickable  false
* password   false
* selected   false
# rect      {"x":707,"y":558,"width":203,"height":452} 代码

```

其中最常用的，可以去定位到元素的一些属性是：

- `text`
- `resourceId`
- `description`
- `clickable`
- 等等

## weditor的log：对应内部各种api请求调用

此处对应weditor内部会显示对应的请求的log：

```

终端 问题 输出 调试控制台 2: Python Debug Console + □ □ ^ ×
[I 210402 22:52:50 web:2246] 200 GET /static/running.svg (::1) 13.29ms
[I 210402 22:52:50 web:2246] 200 GET /static/libs/fontawesome/fonts/fontawesome-webfont.woff2?v=4.7.0 (::1) 17.68ms
[I 210402 22:52:50 web:2246] 200 GET /cdn.jsdelivr.net/bootstrap/3.3.7/fonts/glyphicons-halflings-regular.woff2 (::1)
19.46ms
[I 210402 22:52:50 web:2246] 200 GET /api/v1/version (::1) 3.20ms
[I 210402 22:52:50 web:2246] 101 GET /ws/v1/python (::1) 14.49ms
[I 210402 22:52:50 shell:172] create process pid: 15728
[I 210402 22:52:50 web:2246] 200 GET /unpkg.com/element-ui/lib/theme-chalk/fonts/element-icons.woff (::1) 22.82ms
[W 210402 22:52:51 web:2246] 404 GET /apple-touch-icon-precomposed.png (::1) 2.62ms
[W 210402 22:52:51 web:2246] 404 GET /apple-touch-icon.png (::1) 2.38ms
[I 210402 22:52:51 web:2246] 200 GET /favicon.ico (::1) 5.28ms
[I 210402 22:55:37 web:2246] 200 POST /api/v1/connect (::1) 74.36ms
[I 210402 22:56:04 web:2246] 200 POST /api/v1/connect (::1) 26.35ms
[I 210402 22:56:05 web:2246] 200 GET /api/v2/devices/android%3Aorga4pmzee4ts47t/hierarchy (::1) 1126.95ms
[I 210402 22:56:06 web:2246] 200 GET /api/v2/devices/android%3Aorga4pmzee4ts47t/hierarchy (::1) 756.98ms
[I 210402 22:56:07 web:2246] 200 GET /api/v2/devices/android%3Aorga4pmzee4ts47t/hierarchy (::1) 647.42ms
[I 210402 22:56:07 web:2246] 200 GET /api/v2/devices/android%3Aorga4pmzee4ts47t/hierarchy (::1) 650.48ms
[I 210402 22:56:08 web:2246] 200 GET /api/v2/devices/android%3Aorga4pmzee4ts47t/hierarchy (::1) 736.39ms
[I 210402 22:56:09 web:2246] 200 GET /api/v2/devices/android%3Aorga4pmzee4ts47t/hierarchy (::1) 718.26ms
[I 210402 22:56:10 web:2246] 200 GET /api/v2/devices/android%3Aorga4pmzee4ts47t/hierarchy (::1) 739.85ms
[I 210402 22:56:11 web:2246] 200 GET /api/v2/devices/android%3Aorga4pmzee4ts47t/hierarchy (::1) 944.19ms
[I 210402 22:56:11 web:2246] 200 GET /api/v2/devices/android%3Aorga4pmzee4ts47t/hierarchy (::1) 790.93ms
行 22, 列 54 (已选择623) 空格: 2 UTF-8 LF 纯文本 Rsync: ⓘ ⌂ Prettier ⌂ ⌂
```

```

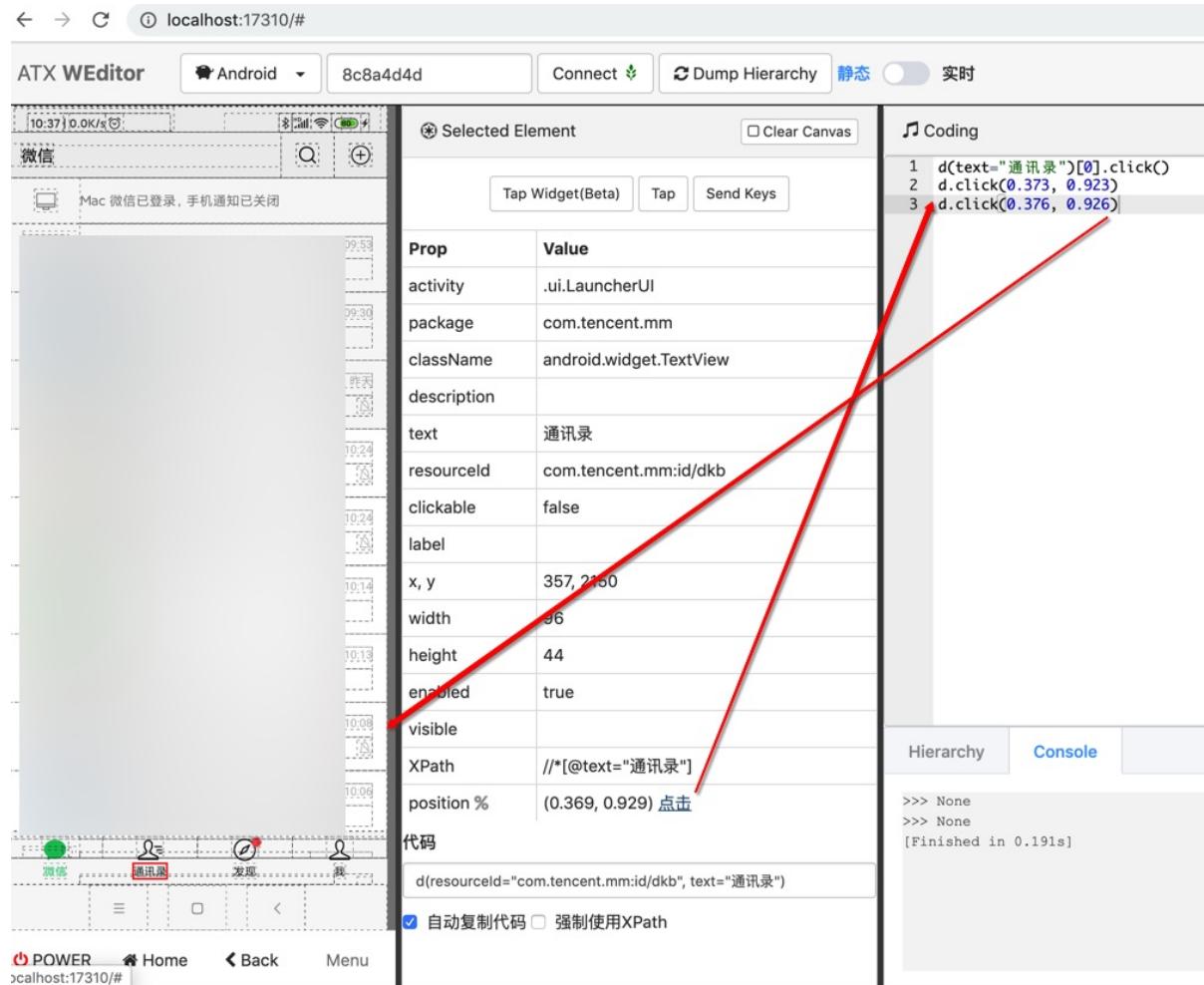
python -m weditor
listening on http://192.168.31.12:17310
[I 210402 22:52:49 web:2246] 200 GET / (::1) 18.51ms
[I 210402 22:52:49 web:2246] 200 GET /cdn.jsdelivr.net/bootstrap/3.3.7/css/bootstrap.min.css (::1) 19.03ms
[I 210402 22:52:49 web:2246] 200 GET /cdn.jsdelivr.net/npm/jstree@3.3.8/dist/themes/default/style.min.css (::1) 33.71ms
[I 210402 22:52:49 web:2246] 200 GET /static/libs/fontawesome/css/font-awesome.min.css (::1) 37.99ms
[I 210402 22:52:49 web:2246] 200 GET /cdn.jsdelivr.net/bootstrap.select/1.12.2/css/bootstrap-select.min.css (::1) 42.20ms
[I 210402 22:52:49 web:2246] 200 GET /unpkg.com/element-ui/lib/theme-chalk/index.css (::1) 52.95ms
[I 210402 22:52:49 web:2246] 200 GET /static/libs/css/buttons.css (::1) 59.13ms
[I 210402 22:52:49 web:2246] 200 GET /static/style.css?v=36c1d231ca96af637a1b43fdeec15aec (::1) 64.98ms
[I 210402 22:52:49 web:2246] 200 GET /cdn.jsdelivr.net/npm/jquery@3.3.1/dist/jquery.min.js (::1) 20.32ms
[I 210402 22:52:50 web:2246] 200 GET /cdn.jsdelivr.net/bootstrap/3.3.7/js/bootstrap.min.js (::1) 81.69ms
[I 210402 22:52:50 web:2246] 200 GET /cdn.jsdelivr.net/bootstrap.select/1.12.2/js/bootstrap-select.min.js (::1) 100.70ms
[I 210402 22:52:50 web:2246] 200 GET /cdn.jsdelivr.net/npm/ace-builds@1.4.12/src-noconflict/ace.min.js (::1) 110.94ms

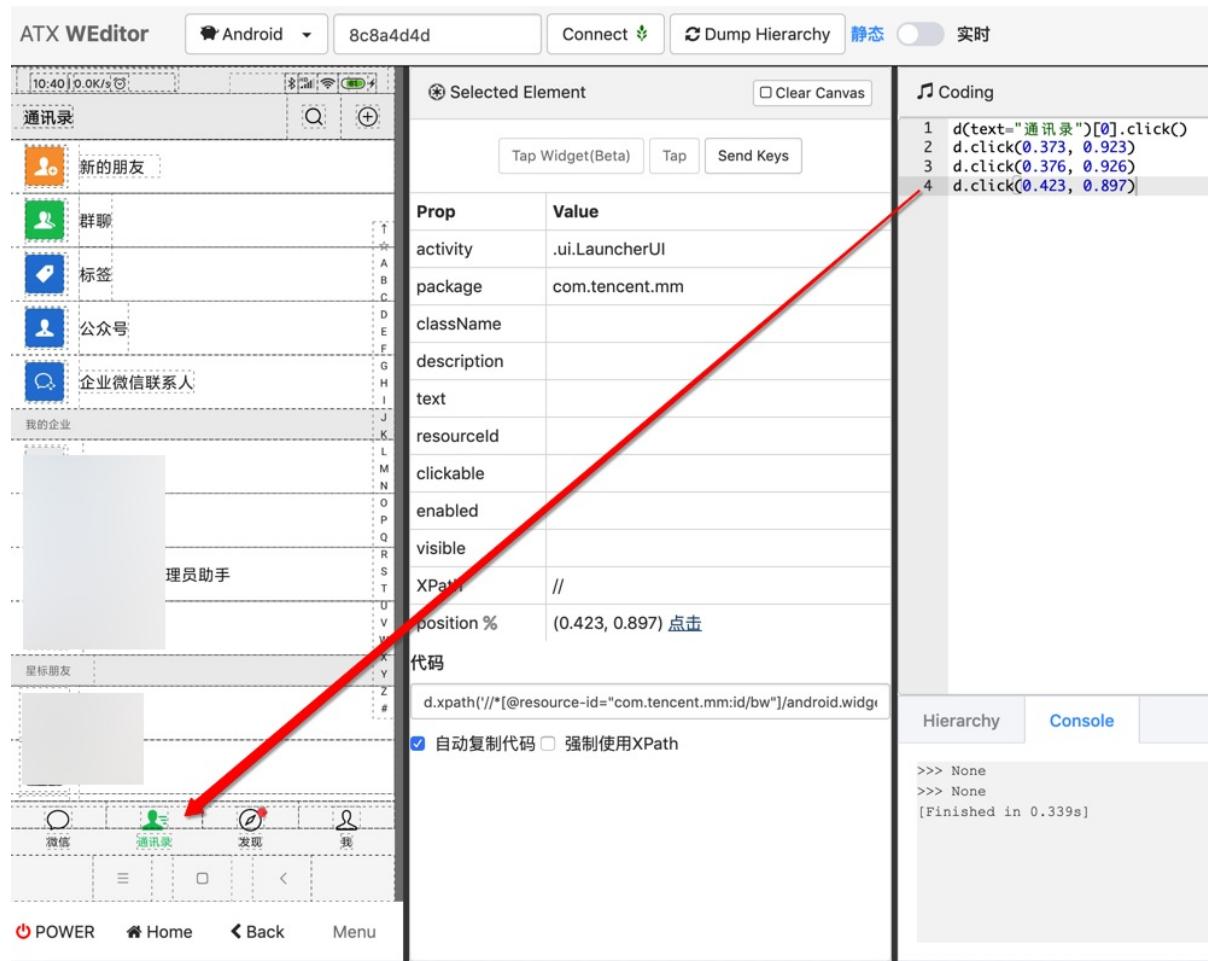
```

```
[I 210402 22:52:50 web:2246] 200 GET /cdn.jsdelivr.net/npm/ace-builds@1.4.12/src-noconflict/mode-python.min.js (::1) 124.62ms
[I 210402 22:52:50 web:2246] 200 GET /cdn.jsdelivr.net/npm/ace-builds@1.4.12/src-noconflict/ext-language_tools.min.js (::1) 135.78ms
[I 210402 22:52:50 web:2246] 200 GET /cdn.jsdelivr.net/npm/ace-builds@1.4.12/src-noconflict/ext-searchbox.min.js (::1) 8.32ms
[I 210402 22:52:50 web:2246] 200 GET /cdn.jsdelivr.net/npm/ace-builds@1.4.12/src-noconflict/theme-monokai.js (::1) 16.38ms
[I 210402 22:52:50 web:2246] 200 GET /cdn.jsdelivr.net/npm/jstree@3.3.8/dist/jstree.min.js (::1) 23.69ms
[I 210402 22:52:50 web:2246] 200 GET /static/libs/vue-2.5.16/vue.js?v cbe2b9b2fb6955decf033515d079e44b (::1) 31.83ms
[I 210402 22:52:50 web:2246] 200 GET /unpkg.com/element-ui/lib/index.js (::1) 50.16ms
[I 210402 22:52:50 web:2246] 200 GET /static/js/common.js?v f1c2d6b24d012c4fc2e22595bd549793 (::1) 53.17ms
[I 210402 22:52:50 web:2246] 200 GET /static/js/index.js?v 1e93d9fb27bafdd173d79477b015322e (::1) 5.14ms
[I 210402 22:52:50 web:2246] 200 GET /static/loading.svg (::1) 7.07ms
[I 210402 22:52:50 web:2246] 200 GET /static/running.svg (::1) 13.29ms
[I 210402 22:52:50 web:2246] 200 GET /static/libs/fontawesome/fonts/fontawesome-webfont.woff2?v 4.7.0 (::1) 17.68ms
[I 210402 22:52:50 web:2246] 200 GET /cdn.jsdelivr.net/bootstrap/3.3.7/fonts/glyphicons-halflings-regular.woff2 (::1) 19.46ms
[I 210402 22:52:50 web:2246] 200 GET /api/v1/version (::1) 3.20ms
[I 210402 22:52:50 web:2246] 101 GET /ws/v1/python (::1) 14.49ms
[I 210402 22:52:50 shell:172] create process pid: 15728
[I 210402 22:52:50 web:2246] 200 GET /unpkg.com/element-ui/lib/theme-chalk/fonts/element-icons.woff (::1) 22.82ms
[W 210402 22:52:51 web:2246] 404 GET /apple-touch-icon-precomposed.png (::1) 2.62ms
[W 210402 22:52:51 web:2246] 404 GET /apple-touch-icon.png (::1) 2.38ms
[I 210402 22:52:51 web:2246] 200 GET /favicon.ico (::1) 5.28ms
[I 210402 22:55:37 web:2246] 200 POST /api/v1/connect (::1) 74.36ms
[I 210402 22:56:04 web:2246] 200 POST /api/v1/connect (::1) 26.35ms
[I 210402 22:56:05 web:2246] 200 GET /api/v2/devices/android%3Aorga4pmzee4ts47t/hierarchy (::1) 1126.95ms
[I 210402 22:56:06 web:2246] 200 GET /api/v2/devices/android%3Aorga4pmzee4ts47t/hierarchy (::1) 756.98ms
...
[I 210402 22:56:39 web:2246] 200 GET /api/v2/devices/android%3Aorga4pmzee4ts47t/hierarchy (::1) 743.71ms
[I 210402 22:56:39 page:204] Serial: android:orga4pmzee4ts47t
[I 210402 22:56:40 web:2246] 200 GET /api/v1/devices/android%3Aorga4pmzee4ts47t/screenshot (::1) 475.57ms
[I 210402 22:56:41 web:2246] 200 GET /api/v2/devices/android%3Aorga4pmzee4ts47t/hierarchy (::1) 946.91ms
```

## Coding中可以调试代码

之前有用过输入并运行代码，用于调试，效果不错：





再比如：

```

d(className="android.view.View")
d(className="android.view.View").count
    
```

选中第一行后，点击 选中运行：

实时 Command+Shift+Enter

Coding

1 d(className="android.view.View")  
2 d(className="android.view.View").count

重置代码 选中运行 停止

Hierarchy Console

```
>>> <uiautomator2.session.UiObject object at 0x1068653a0>
>>> None
[Finished in 0.001s]
```

加上print后

```
print(d(className "android.view.View"))
print(d(className "android.view.View").count)
```

不选中，点击 运行按钮，表示全部运行：

The screenshot shows the weditor interface with the following components:

- Real-time tab:** Shows the word "实时" (Real-time).
- Toolbar:** Includes a "相关文档" dropdown, a "Command+Enter" button, and several action buttons: "重置代码" (Reset Code), "运行" (Run), "选中运行" (Run Selected), and "停止" (Stop).
- Coding Area:** Displays the following Python code:

```
1 print(d(className="android.view.View"))
2 print(d(className="android.view.View").count)
```
- Console Area:** Shows the output of the executed code:

```
<uiautomator2.session.UiObject object at 0x10684e340>
7
>>> None
[Finished in 0.064s]
```
- Bottom Navigation:** Features tabs for "Hierarchy" and "Console".

可以实时调试，很方便。

详见：

【未解决】自动抓包工具抓包公众号买单吧某个元素通过class+instance定位不到

【已解决】uiautomator2用click点击微信中的通讯录不起作用

Hierarchy支持有限的搜索

对于xml中的节点：

```
<node NAF="true" index="0" text="" resource-id="com.tencent.mm:id/pq" class="android.view.View" package="com.tencent.mm" content-desc="" checkable="false" checked="false" clickable="true" enabled="true" focusable="true" focused="false" scrollable="false" long-clickable="true" password="false" selected="false" visible-to-user="true" bounds="[156,1522][912,2027]" />
```

想要去WEitor中

搜id值，即搜 com.tencent.mm:id/pq，结果找不到

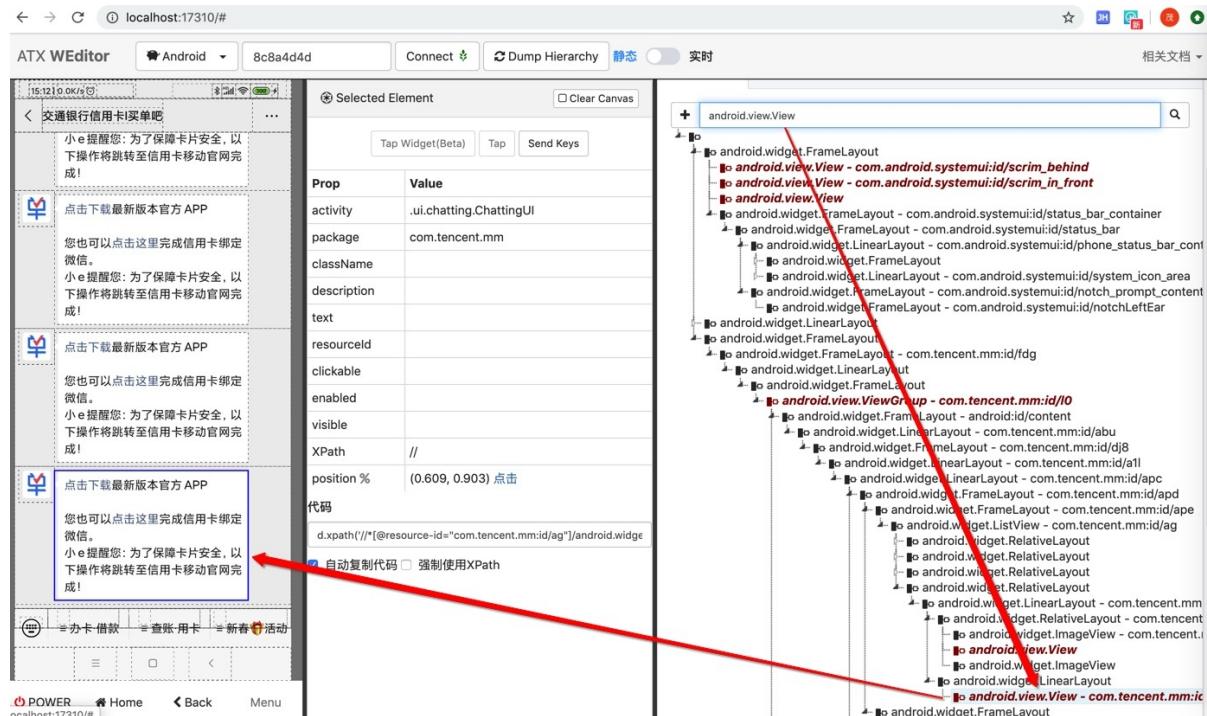
搜pq，也搜不到

后来发现，只能搜：当前显示出来的内容，即节点的class的类型

比如： android.view.View

是可以搜出并深红色高亮显示的对应节点的

然后才找到此处对应节点：



详见：

**【已解决】用weditor实时查看安卓当前页面中的xml源码**

**【已解决】Mac中安装uiautomator2的UI界面工具：weditor**

查看当前的app的包名

举例：

用weditor可以查看到当前的app的包名：

- 小米的 内置浏览器 包名： `com.android.browser`

◦

- Edge包名： `com.microsoft.emmx`

- - Firefox包名: `org.mozilla.firefox`

- - UC浏览器包名: com.UCMobile

- - QQ浏览器包名: com.tencent.mtt

◦

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2021-07-18 09:59:23

## adb

此处整理uiautomator2开发期间，用到的和 adb 相关的东西。

### 解锁屏幕

安卓手机，华为的DIG-AL00，想要从锁屏界面，进入（等待输入密码的）解锁界面，可以用：

```
adb -s DWH9X17124W03779 shell input swipe 300 300 500 1000 100
```

或：

```
adb -s DWH9X17124W03779 shell input touchscreen swipe 300 300 500 1000 100
```

其中：

- DWH9X17124W03779：是手机的序列号
- 300 300 500 1000 100：
  - 300 300 500 1000：是屏幕坐标：X，Y，宽，高
  - 100：滑动时间，单位毫秒

即可实现进入解锁界面。

输入文字（密码）：

```
adb -s DWH9X17124W03779 shell input text 1234
```

用于解锁手机。

特殊：

- 华为畅享6S DIG-AL00：不支持
  - 相关信息
    - android版本:6.0
    - 系统：EMUI 4.1
  - 原因：估计是系统问题
  - 解决办法：无法解决

详见：

【无法解决】adb发送密码无法解锁安卓手机屏幕

### adb shell中的am start命令

android的adb调试工具，有个shell，可以执行很多命令。

其内部都是调用对应的子工具去处理具体功能的。

此处相关的有：

- 调用 Activity 管理器 ( am )
  - Activity Manager
- 调用软件包管理器 ( pm )
- 调用设备政策管理器 ( dpm )

其中am的解释是：

在 adb shell 中，您可以使用 Activity 管理器 (am) 工具发出命令以执行各种系统操作，如启动某项 Activity、强行停止某个进程、广播 intent、修改设备屏幕属性，等等。

在 shell 中，相应的语法为：

```
am command
```

您也可以直接从 adb 发出 Activity 管理器命令，无需进入远程 shell。例如：

```
adb shell am start -a android.intent.action.VIEW
```

具体参数含义解释：

command的语法=可用的 Activity 管理器命令

有很多，其中的start的语法是：

```
start [options] intent
```

- options=选项，包括：
  - -D：启用调试。
  - -W：等待启动完成。
  - --start-profiler file：启动分析器并将结果发送到 file。
  - -P file：类似于 --start-profiler，但当应用进入空闲状态时分析停止。
  - -R count：重复启动 Activity count 次。在每次重复前，将完成顶层 Activity。
  - -S：启动 Activity 前强行停止目标应用。
  - --opengl-trace：启用对 OpenGL 函数的跟踪。
  - --user user\_id | current：指定要作为哪个用户运行；如果未指定，则作为当前用户运行。
- intent：启动 intent 指定的 Activity。
  - (主要) 语法是：
    - -a action
      - 指定 intent 操作，例如 android.intent.action.VIEW（只能声明一次）。
    - -d data\_uri
      - 指定 intent 数据 URI，例如 content://contacts/people/1（只能声明一次）。
    - -t mime\_type
      - 指定 intent MIME 类型，例如 image/png（只能声明一次）。

- -c category
  - 指定 intent 类别, 例如 android.intent.category.APP\_CONTACTS。
- -n component
  - 指定带有软件包名称前缀的组件名称以创建显式 intent, 例如 com.example.app/.ExampleActivity。
- -f flags
  - 将标记添加到 setFlags() 支持的 intent。
- --esn extra\_key
  - 添加一个空 extra。URI intent 不支持此选项。
- -e | --es extra\_key extra\_string\_value
  - 将字符串数据作为键值对添加进来。
- --ez extra\_key extra\_boolean\_value
  - 将布尔型数据作为键值对添加进来。
- --ei extra\_key extra\_int\_value
  - 将整型数据作为键值对添加进来。
- --el extra\_key extra\_long\_value
  - 将长整型数据作为键值对添加进来。
- --ef extra\_key extra\_float\_value
  - 将浮点型数据作为键值对添加进来。
- --eu extra\_key extra\_uri\_value
  - 将 URI 数据作为键值对添加进来。
- --ecn extra\_key extra\_component\_name\_value
  - 添加组件名称, 该名称作为 ComponentName 对象进行转换和传递。
- --eia extra\_key extra\_int\_value[,extra\_int\_value...]
  - 添加整数数组。
- --ela extra\_key extra\_long\_value[,extra\_long\_value...]
  - 添加长整数数组。
- --efa extra\_key extra\_float\_value[,extra\_float\_value...]
  - 添加浮点数数组。

此处的：

- am start -a android.intent.action.MAIN -c android.intent.category.LAUNCHER -n com.tencent.mm/.ui.LauncherUI
  - am start : 启动
  - -a android.intent.action.MAIN : intent的动作是 android.intent.action.MAIN
  - -c android.intent.category.LAUNCHER : intent类别是 android.intent.category.LAUNCHER
  - -n com.tencent.mm/.ui.LauncherUI
    - 要启动的app=包名: com.tencent.mm
      - 也就是微信
    - 要启动的activity=界面=页面: .ui.LauncherUI
      - 也就是微信的主页面



## android-uiautomator-server

<https://github.com/openatx/android-uiautomator-server>

其发布的

<https://github.com/openatx/android-uiautomator-server/releases>

是2个apk:

- app-uiautomator-test.apk
- app-uiautomator.apk

其具体编译过程是：

```
$ ./gradlew build  
$ ./gradlew packageDebugAndroidTest
```

会生成apk，而最终的2个apk是mv生成的。

详见：.travis.yml 中的：

```
script:  
  - "./gradlew build"  
  - "./gradlew packageDebugAndroidTest"  
before_deploy:  
  - mv app/build/outputs/apk/debug/app-debug.apk app/build/outputs/apk/app-uiautomator.apk  
  - mv app/build/outputs/apk/androidTest/debug/app-debug-androidTest.apk app/build/outputs/apk/app-uiautomator-test.apk
```

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2020-12-11 21:39:52

## uiautomator

android官网的工具：uiautomator

主页：[uiautomator | Android Developers](#)

说了具体用法：

```
adb shell uiautomator dump
```

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2020-08-09 10:14:28

## 常见问题

此处整理出uiautomator2开发期间，遇到的一些常见问题及其解决办法。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2021-07-18 09:59:23

## 输入文字

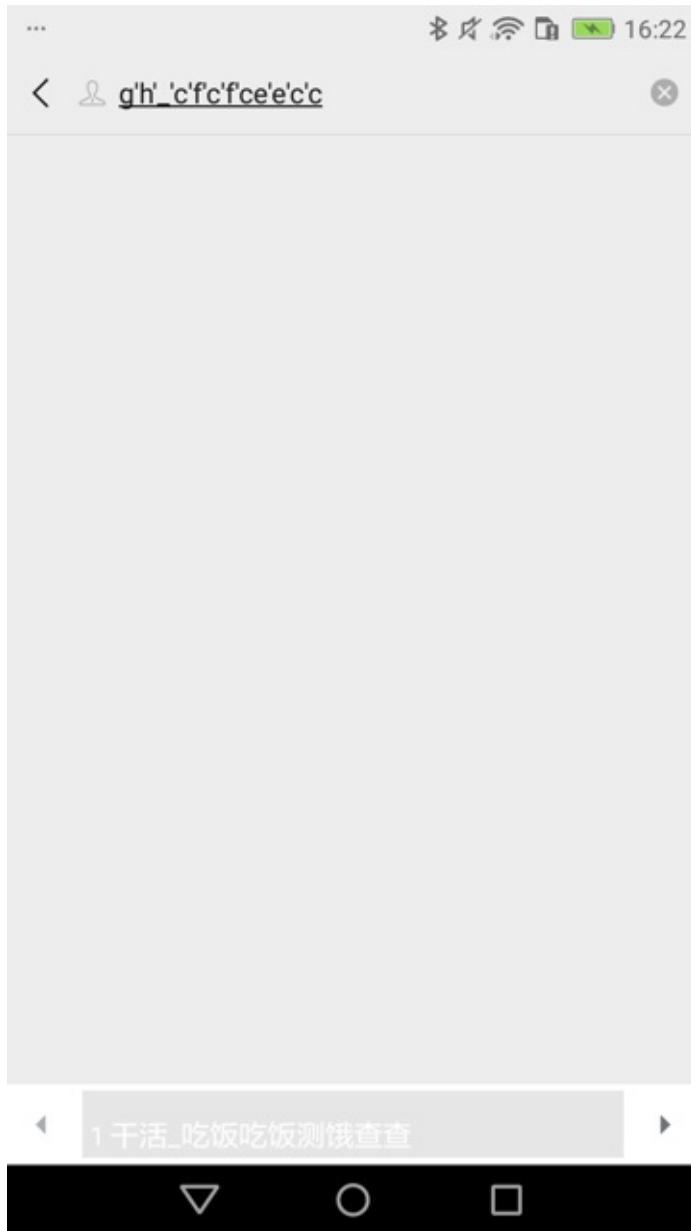
### 输入文字的两种方式

对于输入文字，发现之前的可以工作的代码：

```
self.driver(text locator["text"]).set_text(text,timeout WaitFind)
```

会出现：无法完整输入内容

具体现象：中文输入法中，输入了字母，但是丢失了数字的效果：



且输入法此时已经也被换了（换成了 FastInputIME 或 系统自带（华为Swype） 输入法了）

注：

```
self.driver(text_locator["text"]).set_text(text, timeout=WaitFind)
```

内部是调用的uiautomator2的session的set\_text：

文件： /Users/limao/.pyenv/versions/3.8.0/lib/python3.8/site-packages/uiautomator2/session.py

```
def set_text(self, text, timeout=None):
    self.must_wait(timeout=timeout)
    if not text:
        return self.jsonrpc.clearTextField(self.selector)
    else:
        return self.jsonrpc.setText(self.selector, text)
```

(除了额外支持timeout参数外)

而换用另外的：

xpath 的 set\_text

```
searchElementSelector = self.driver.xpath(searchKeyText)
searchElementSelector.set_text(text)
```

内部调用的：

文件： /Users/limao/.pyenv/versions/3.8.0/lib/python3.8/site-packages/uiautomator2/xpath.py

```
def set_text(self, text: str = ""):
    el = self.get()
    self._parent.send_text() # switch ime
    el.click() # focus input-area
    self._parent.send_text(text)
```

send\_keys

```
self.driver.send_keys(text)
self.driver.set_fastinputime(False) # 关掉FastInputIME输入法，切换回系统默认输入法（此处华为手机默认输入法是华为Swype输入法）
```

其中，是否加上 打开FastInputIME

```
self.driver.set_fastinputime(True) # # 切换成FastInputIME输入法
self.driver.send_keys(text)
self.driver.set_fastinputime(False) # 关掉FastInputIME输入法，切换回系统默认输入法（此处华为手机默认输入法是华为Swype输入法）
```

经测试，感觉没区别。

结果都是：

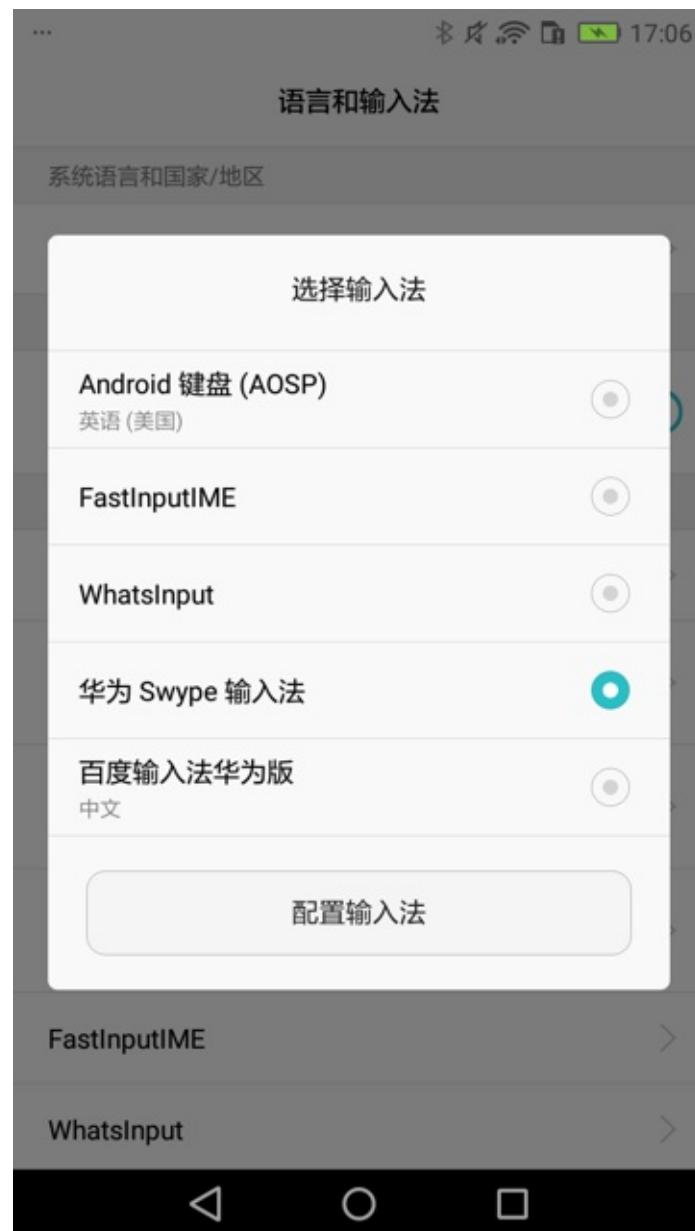
- 可以成功输入文字
  - 此处的：gh\_cfcfcee032cc
- 但是输入法会被切换掉
  - 我之前设置的是：百度的输入法

```
* I[android_input_method_baidu](../assets/img/android_input_method_baidu.png)
```

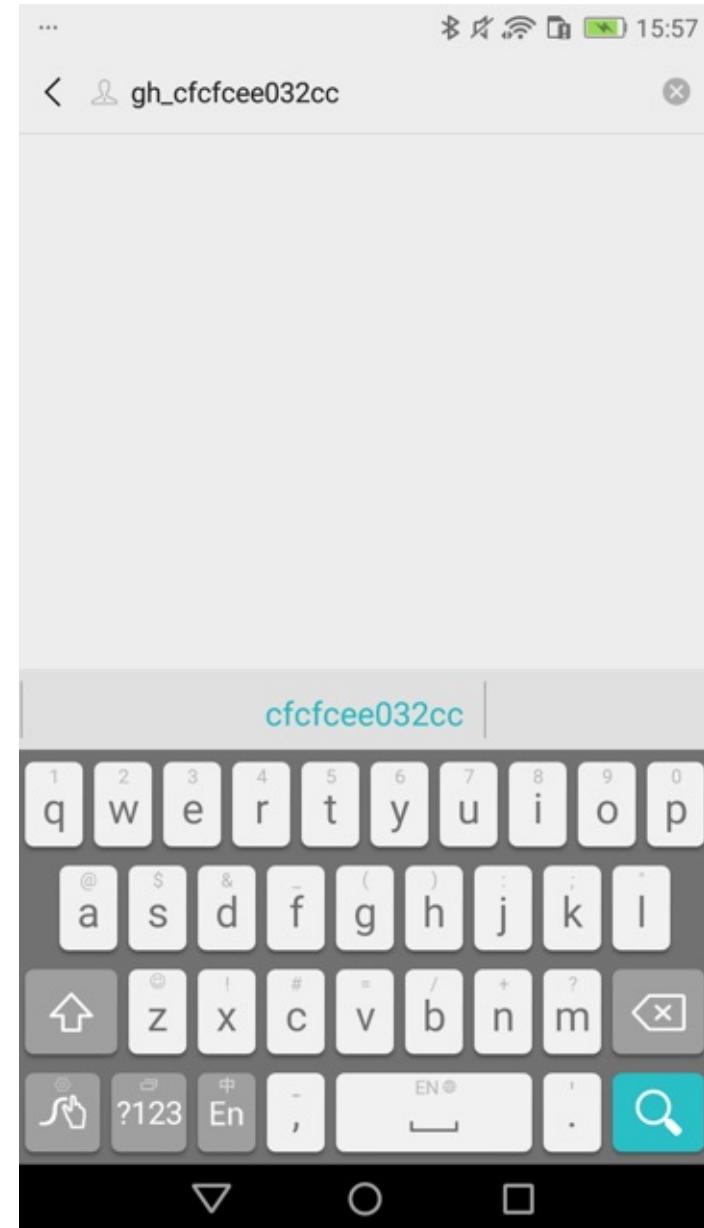
- 对应着，输入文字之前，应该是



- 会被换成：当前系统默认自带输入法
  - 当前系统是：华为的畅享6S手机 DIG-AL00
  - 自带输入法是：华为Swype输入法



- 效果是：



结论：

- 基本上实现了自己的：要输入文字的目的
- 但是：却把之前设置的（百度）输入法切换成系统的（华为）输入法了。
  - 问题不大，但是很不爽
    - 但是没办法改变和保留原有输入法

set\_text导致输入法切换，需要恢复

最终整理出函数：

```
def selectorSetText(u2Dev, curXPathSelector, inputText):  
    selectorSetTextResp = curXPathSelector.set_text(inputText)  
    logging.info("selectorSetTextResp=%s", selectorSetTextResp) # selectorSetTextResp=None  
    # 在set_text后，输入法会变成FastInputIME输入法
```

```
# 用下面代码可以实现：关掉FastInputIME输入法，切换回系统默认输入法
u2Dev.set_fastinput_ime(False)
```

## 用set\_text输入字符串：小米安全键盘 影响输入，可以考虑禁止掉

代码本身：

```
passwordStr = "请输入密码"
passwordXPath = """//android.widget.EditText[@text="%s" and @index="2" and @clickable="true"]""" % passwordStr
passwordSelector = u2Dev.xpath(passwordXPath)
if passwordSelector.exists:
    logging.info("Found %s", passwordStr)
    # pwdClickResp = passwordSelector.click()
    # logging.debug("pwdClickResp=%s", pwdClickResp)
    # doScreenshot(u2Dev)
    selectorSetText(u2Dev, passwordSelector, Vivo_Password)

def selectorSetText(u2Dev, curXPathSelector, inputText):
    selectorSetTextResp = curXPathSelector.set_text(inputText)
    logging.info("selectorSetTextResp=%s", selectorSetTextResp) # selectorSetTextResp=None
    doScreenshot(u2Dev)
    # 在set_text后，输入法会变成FastInputIME输入法
    # 用下面代码可以实现：关掉FastInputIME输入法，切换回系统默认输入法
    u2Dev.set_fastinput_ime(False)
```

是可以输入密码=字符串的

但是

- 之前开启了：小米安全键盘
  - 导致：输入不顺利
    - 小米安全键盘 会弹出显示 消失掉，多次之后
    - （等待1, 2秒后）触发异常：
      - /Users/limao/dev/xxx/crawler/appAutoCrawler/AppCrawler/venv/lib/python3.8/site-packages/uiautomator2/\_\_init\_\_.py:1646: Warning: set FastInputIME failed.
      - use "d(focused=True).set\_text instead"
      - warnings.warn(
    - 最终才能输入密码
- 解决办法：关闭 小米安全键盘
  - 步骤：
    - 系统设置-》更多设置-》语言与输入法-》安全键盘-》取消勾选：开启安全键盘



## 安全键盘

开启安全键盘



## 以为输入框set\_text输入文字无效

某次调试代码，以为是：uiautomator的UiObject或XPathSelector的set\_text不生效问题

其实不是这个原因，而是点击了页面元素本身后，应该进去新的界面（输入法弹框界面，用于输入内容）

所以应该把代码：

```
inputUiObj = d(resourceId:"com.android.browser:id/b4h", className="android.widget.TextView")
)
inputUiObj.setText(BaiduHomeUrl)
```

换成：

```
inputUiObj = d(resourceId:"com.android.browser:id/b4h", className="android.widget.TextView")
)
inputUiObj.click()
```

或：

```
inputXPathSelector = d.xpath("// android.widget.TextView[@resource-id='com.android.browser:id/b4h']")
inputXPathSelector.click()
```

去触发新界面，即可。

-» 后续再去定位搜索框，输入内容（百度首页地址），进入百度首页

crifan.com，使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2021-07-18 09:59:23

## xml源码不是最新的

之前遇到过一个很特殊的问题（bug），用dump\_hierarchy去获取页面xml源码，但是获取的xml不是当前手机中显示的最新的页面的源码，而是旧的。

比如：之前某次情况是，当前是百度首页，但是输入了内容，触发了搜索后，页面内容已变化

但是用代码：

```
d.dump_hierarchy(compressed=False, pretty=False)
```

输出的安卓的页面的xml源码却不是最新的

原因：是之前遇到的，uiautomator2的bug

解决办法：重启uiautomator2的服务

代码：

```
d.service("uiautomator").stop()
d.service("uiautomator").start()
time.sleep(1)
```

即可规避问题，然后获取到xml源码，就是最新的了，包含了百度搜索的结果了。

crifan.com，使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2021-07-18 09:59:23

## 重启服务

当偶尔遇到uiautomator2本身出问题，而后台服务停止或者异常时，可以去重启uiautomator2的服务。

经研究，总结出相关代码：

```
def u2ServiceRestart(self):
    """ restart uiautomator2 service """
    # self.driver.reset_uiautomator()
    # self.driver.service.stop()
    # self.driver.service.start()
    self.driver.service("uiautomator").stop()
    self.driver.service("uiautomator").start()
    # self.driver.uiautomator.stop()
    # self.driver.uiautomator.start()
    time.sleep(1)
```

调用举例：

```
def wait_AccountSearched(self, account):
    ...
    if self.isAndroid:
        # Note1:
        # for 华为畅享6S android 6: following can not get latest page xml after search weix
        in public account
        # here reset uiautomator2 is workaround for later d.dump_hierarchy get latest pag
        e xml
        # Note2: 小米9 Android 10 / 红米 Note8 Pro Android 9: Not need reset uiautomator2 a
        nymore
        isNeedRestartU2 = False

        curAndroidVersionFloat = self.getAndroidVersion()
        ANDROID_VERSION_NEED_RESTART_U2 = 7.0
        if curAndroidVersionFloat <= ANDROID_VERSION_NEED_RESTART_U2:
            isNeedRestartU2 = True

        if isNeedRestartU2:
            self.u2ServiceRestart()
```

## NAF

此处所用安卓手机： 华为畅享6S DIG-AL00



对于最新版的 v7.0.8 的微信，公众号搜索结果的页面，去导出源码，发现：

- `uiautomator2` 中用代码：`self.driver.dump_hierarchy()`
  - 只能导出部分页面的源码
    - 其中红框内的源码无法导出



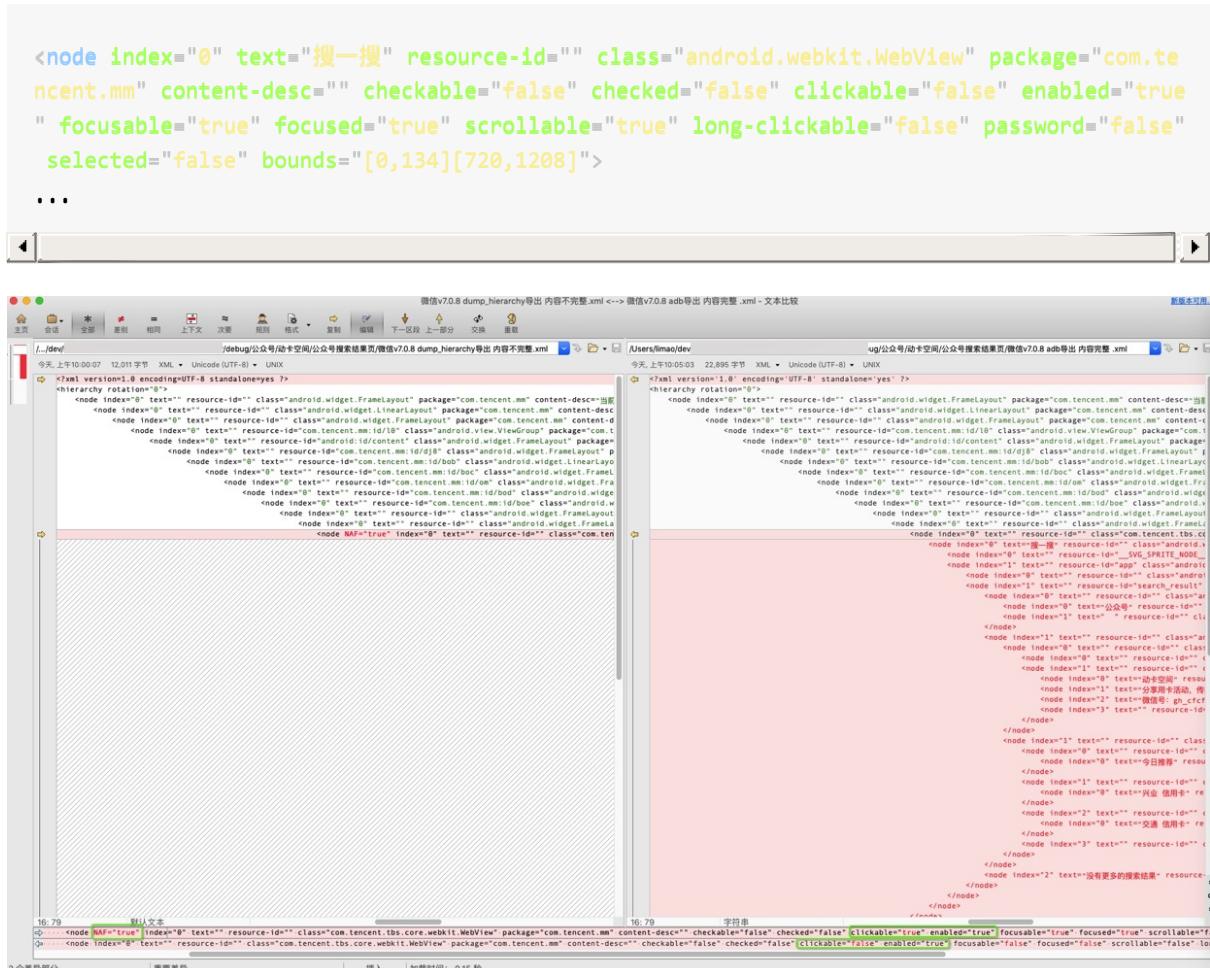
- Mac中终端运行adb命令: `adb shell uiautomator dump`
  - 能导出完整页面的源码

核心代码差异是:

```
<node NAF="true" index="0" text="" resource-id="" class="com.tencent.tbs.core.webkit.WebView" package="com.tencent.mm" content-desc="" checkable="false" checked="false" clickable="true" enabled="true" focusable="true" focused="true" scrollable="false" long-clickable="true" password="false" selected="false" bounds="[0,134][720,1208]" />
```

和

```
<node index="0" text="" resource-id="" class="com.tencent.tbs.core.webkit.WebView" package="com.tencent.mm" content-desc="" checkable="false" checked="false" clickable="false" enabled="true" focusable="false" focused="false" scrollable="false" long-clickable="false" password="false" selected="false" bounds="[0,134][720,1208]">
```



另外：

之前旧版本 v6.7.3的微信，是可以正常导出的。

所以去研究：

微信版本升级前后的页面源码的变化：

- 升级前 = 微信 v6.7.3

- 2018微信v6.7.3老旧历史版本安装包官方免费下载\_豌豆荚
- 页面源码：

```

<node index="0" text="" resource-id="" class="android.webkit.WebView" package="com.tencent.mm" content-desc="" checkable="false" checked="false" clickable="false" enabled="true" focusable="true" focused="false" scrollable="false" password="false" selected="false" bounds="[0,144][720,1208]">
<node index="0" text="" resource-id="" class="android.webkit.WebView" package="com.tencent.mm" content-desc="搜一搜" checkable="false" checked="false" clickable="false" enabled="true" focusable="true" focused="true" scrollable="true" long-clickable="false" password="false" selected="false" bounds="[0,144][720,1208]">

<node index="0" text="" resource-id="__SVG_SPRITE_NODE__" class="android.widget.ImageView" package="com.tencent.mm" content-desc="" checkable="false" checked="false" clickable="false" enabled="true" focusable="false" focused="false" scrollable="false" bounds="[-1,-1][1,-1]">

```

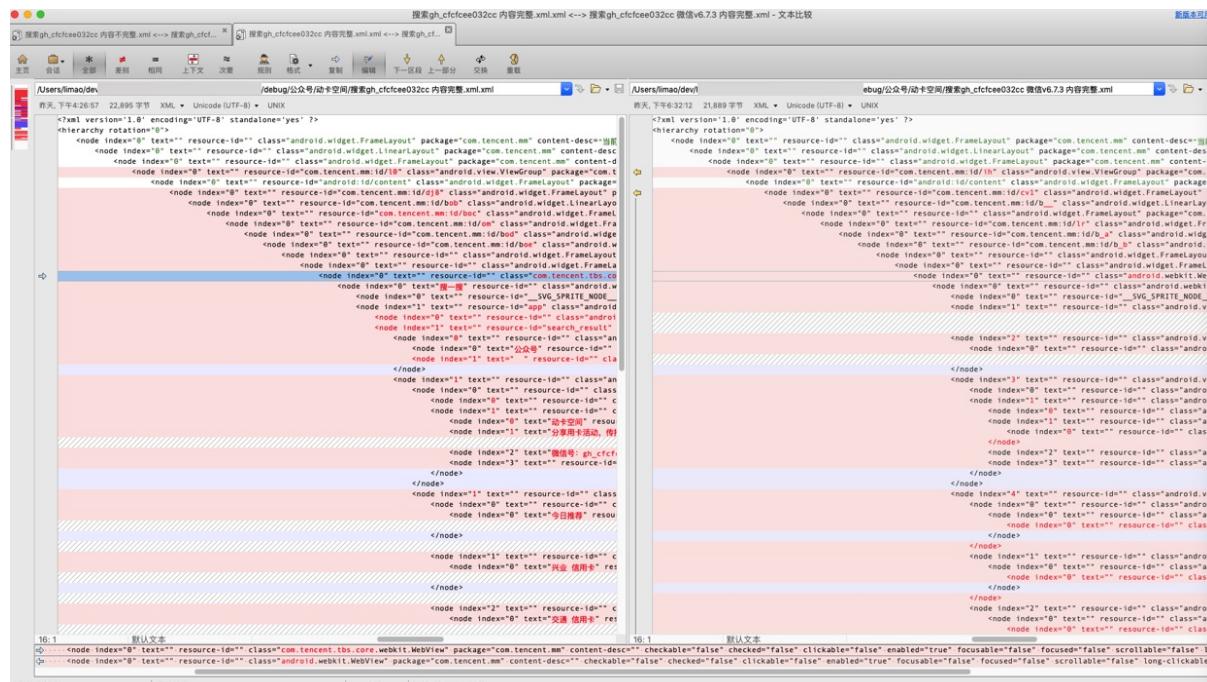


- 升级后 = 微信 v7.0.8

- 2019微信v7.0.8老旧历史版本安装包官方免费下载\_豌豆荚
- 页面源码：



对比的区别：



主要是class的不同：

- v6.7.3 : class="android.webkit.WebView"
- v7.0.8 : class="com.tencent.tbs.core.webkit.WebView"

而 tbs = 腾讯浏览器服务 = X5浏览器 = X5内核

即，新版本微信的浏览器的内核，用上了新的X5内核

- 什么是NAF

- = Not Accessibility Friendly
  - 直译：不可方便地访问（的节点）
    - Accessibility = 可访问性 = 可及性
      - 与之相对的是：（元素节点） Accessible 可访问
        - 其他普通的节点都是属于可访问的
        - 被谁访问：被其他的工具或软件等读取和操作
          - 其他的工具和软件：用来查看和研究Android页面源码的工具
            - 比如
              - android 自带的 uiautomatorviewer
                - 其还支持选项 "Toggle NAF Nodes"，打开后，可以查看到 NAF的节点
                  - 默认不能查看到NAF节点
                - 详见：
                  - [UI Testing | Android Developers](#)
    - Accessibility可访问性：主要指的是描述内容 content description 和 text 文本
      - 只有描述内容或文本有内容
        - 普通用户，才能从页面上才能看到该元素
          - 否则对于普通用户就看不到该元素了，也就没太大意义了
            - 或许可以被视为不可见元素了
    - 节点=元素=xml中的节点=某个UI控件=Android程序中页面上的某个控件
      - =Android的页面源码=xml代码
  - 判定NAF的逻辑=如何判定一个节点是NAF
    - 根据上面的代码中的 !nafExcludedClass(node) && !nafCheck(node) 可以看出：
      - 先判断节点的类型
        - !nafExcludedClass(node)：是属于那些可能被当做NAF节点的类型
          - 哪些节点，可能会被当做NAF节点呢？
            - class="xxx" 中 xxx，即类名不在 NAF\_EXCLUDED\_CLASSES 范围内的
              - 而 NAF\_EXCLUDED\_CLASSES 包括哪些呢？，包括如下：
                - android.widget.GridLayout
                - android.widget.GridView
                - android.widget.ListView
                - android.widget.TableLayout
            - 可见：除了上面4种节点，其他类型的节点，（只要符合特定条件）都可能被判定为NAF
        - 再判断节点的内容是否符合条件

- !nafCheck(node) :
  - 是可点击的
    - XML代码中: `clickable="true"`
  - 是已启用的=是有效的
    - XML代码中: `enabled="true"`
  - 描述内容是空的
    - XML代码中: `content-desc=""`
  - 文本是空的
    - XML代码中: `text=""`
- 如果上述2个条件都满足则判定是：NAF节点
  - 输出的节点中，会加上：`NAF="true"`
  - 这类节点，往往 `resource-id` 也是空
    - 典型的XML源码：
 

```
<node NAF="true" ... text="" resource-id="" class="com.tencent.tbs.core.webkit.WebView" package="com.tencent.mm" content-desc="" ... clickable="true" enabled="true" ... />
```
- 思考：
  - 为何对于：可点击的、已启用的，但是描述内容是空的、文本是空的 节点，被当做 NAF，认为不能被访问到呢？
    - 因为，这类节点，从Android的界面上，往往是看不到的，但是却又能被点击，所以基本上处于不可用状态
      - 所以（代码的作者）认为这类节点，属于（从设计角度来说，就是故意）不想被普通用户看到，接触到
        - 所以被判定为NAF，不应该被访问到
          - 在导出页面源码时，被忽略掉，不导出NAF节点
  - 为何上述4种节点：`GridLayout`，`GridView`，`ListView`，`TableLayout`，不会被当做 NAF呢？
    - 因为：满足了前面的 可点击的、已启用的，但是描述内容是空的、文本是空的 节点
      - 如果是本身属于（Android系统自带的）列表、表格等类型的节点，
        - 看起来就是：属于正常的节点了，因为这类节点，本身是可以没有描述内容，文本是空的
          - 而列表、表格等节点，就是Android中的：`GridLayout`, `GridView`, `ListView`, `TableLayout`等类型的节点
            - 当然，作者自己也说了，这4个类型，未必完整
              - 理论上，你也可以把其他的，合理的节点类型加到这个
                - `NAF_EXCLUDED_CLASSES` =不应该被认为是NAF的节点的类型中

详见：

【已解决】uiautomator2中导出页面源码中NAF是什么意思

而关于NAF如何规避解决，详见：

【未解决】如何确保uiautomator2的dump\_hierarchy能导出页面中NAF的元素节点

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2021-07-18 09:59:23

## long\_click不工作

权限问题导致long\_click不工作

之前小米9中用long\_click

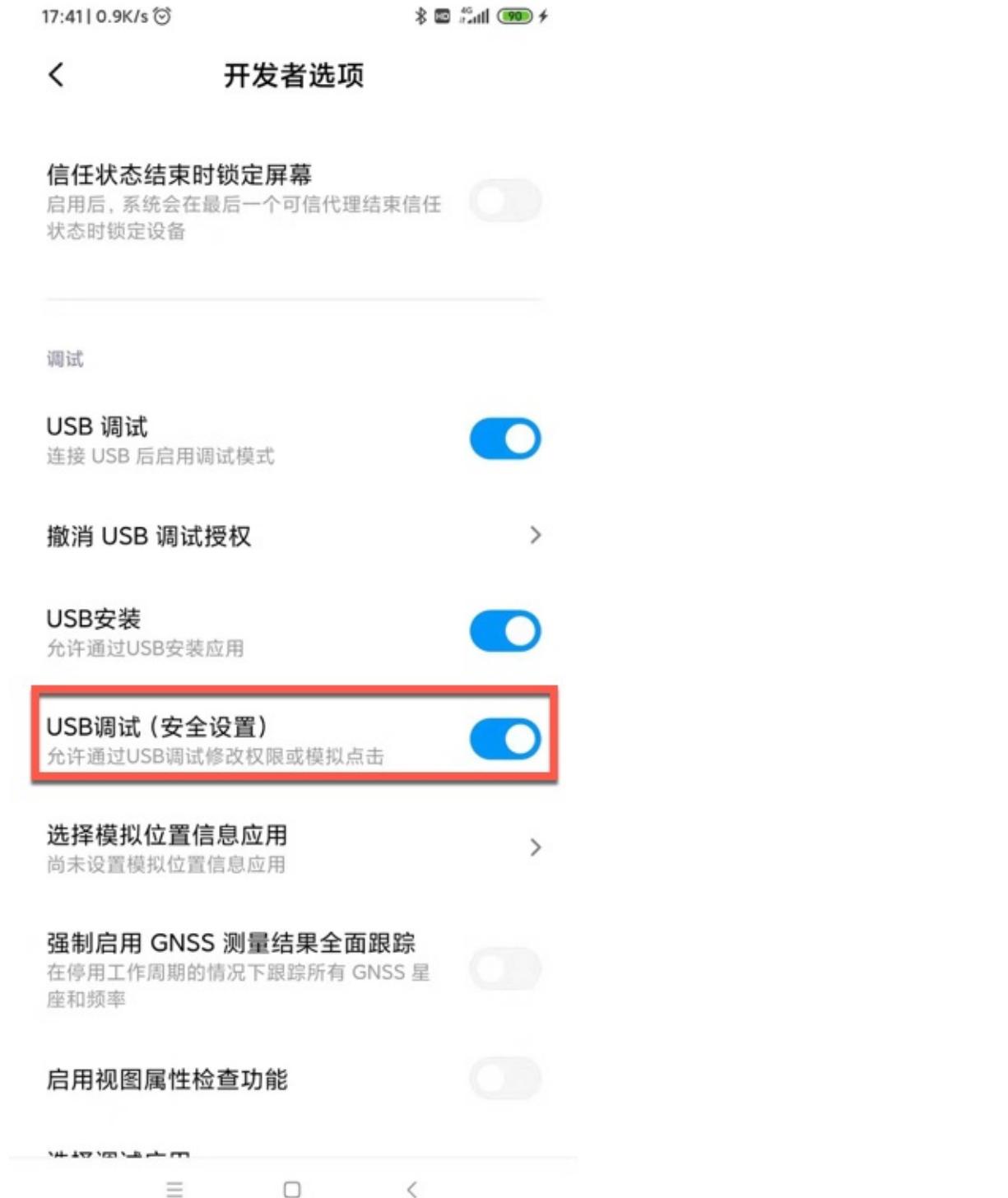
```
self.driver(text locator["text"]).click(timeout WaitFind)
```

报错：

```
viautomator2.exceptions.JsonRpcError: 0 Unknown error: Injecting to another application requires INJECT_EVENTS permission data: {'exceptionTypeName': 'java.lang.SecurityException', 'message': 'Injecting to another application requires INJECT_EVENTS permission'}, method: click
```

即： INJECT\_EVENTS 问题=权限问题

解决办法：去开启权限 USB调试（安全设置） -> 允许通过USB调试修改权限或模拟点击



注：期间会3次提醒你

- 因为这个权限很重要
  - 如果随便给了其他坏的应用
    - 可能会滥用，而导致你手机被恶意操控
    - 所以多次提醒你确认
    - 自己此处是调试手机，自动抓包，所以没问题，是打算开启此权限

新: 2021-07-18 09:59:23

## 后台服务已被杀掉

如果uiautomator2过段时间 偶尔 不定期 被杀掉 导致服务需要重启，则很可能是 后台服务被杀掉了，需要设置 允许后台运行

即把： ATX 、 com.github.uiautomator.test 的 应用智能省电， 改为：无限制

- ATX

◦

- - com.github.uiautomator.test

◦

◦

其他还有类似的：

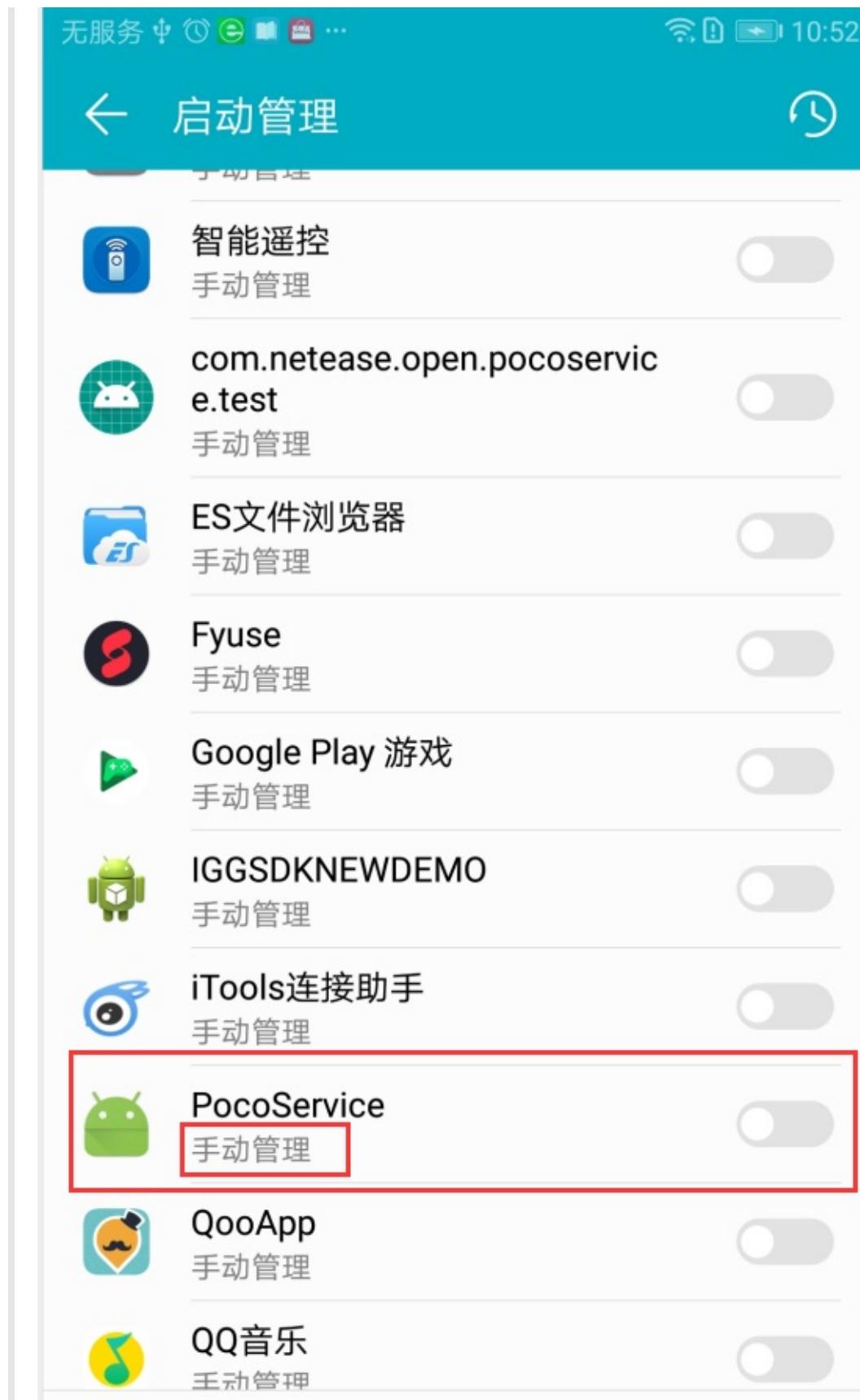
在设置中的 后台高耗电 中：

- 允许后台运行： ATX和com.github.uiautomator.test

◦

另外内部用了 `uiautomator2` 的网易的 `AirTest`，也是类似逻辑：

5.如果华为手机出现poco在启动后十几秒内自动断开的话，可以检查一下手机管家的版本号是否大于8.0，如果是的话，就在手机管家->启动管理里面，找到pocoservice，勾选允许自启动和允许后台活动





## 源码分析

折腾uiautomator2期间，分析了其中部分源码。现把过程整理如下供参考。

**uiautomator-server中最终的底层实现dumpWindowHierarchy的处理返回页面数据的逻辑**

先启动了底层的jsonrpc的服务，监听发送过来的，要执行的动作

文件： app/src/androidTest/java/com/github/uiautomator/stub/Stub.java

```
import com.github.uiautomator.JsonRpcServer;

public class Stub {
    ...
    int PORT = 9008;
    AutomatorHttpServer server = new AutomatorHttpServer(PORT);

    @Before
    public void setUp() throws Exception {
        launchService();
        JsonRpcServer jrs = new JsonRpcServer(new ObjectMapper(), new AutomatorServiceImpl(),
        AutomatorService.class);
        ...
        server.route("/jsonrpc/0", jrs);
        server.start();
    }
}
```

其中对于launchService：

```
private void launchService() throws RemoteException {
    UiDevice device = UiDevice.getInstance(InstrumentationRegistry.getInstrumentation());
    Context context = InstrumentationRegistry.getContext();
    device.wakeUp();

    // Wait for launcher
    String launcherPackage = device.getLauncherPackageName();
    Boolean ready = device.wait(Until.hasObject(By.pkg(launcherPackage).depth(0)), LAU
NCH_TIMEOUT);
    if (ready) {
        Log.i(TAG, "Wait for launcher timeout");
        return;
    }
}
```

```

        Log.d("Launch service");
        startMonitorService(context);
    }

    private void startMonitorService(Context context) {
        Intent intent = new Intent("com.github.uiautomator.ACTION_START");
        intent.setPackage("com.github.uiautomator"); // fix error: Service Intent must be
explicit
        context.startService(intent);
    }

```

去启动了 `com.github.uiautomator`，应该就是在后台运行的uiautomator的服务了。

而前面的 `JsonRpcServer` 的 `jrs`，则是：

- 负责监听 `/jsonrpc/0`
  - 对应着之前 `uiautomator2` 中发送过来的请求
    - Shell\$ curl -X POST -d 'b'{"jsonrpc": "2.0", "id": "1f056baf5d6b2ea2cb7e546efb7cd64f", "method": "dumpWindowHierarchy", "params": [true, null]}' http://127.0.0.1:64445/jsonrpc/0
    - 中的 `jsonrpc/0`
- 其具体实现的类是 `AutomatorServiceImpl` 的 `AutomatorService`
  - 下面就来介绍 `AutomatorServiceImpl`

文件： `app/src/androidTest/java/com/github/uiautomator/stub/AutomatorServiceImpl.java`

```

public class AutomatorServiceImpl implements AutomatorService {

    /**
     * It's to test if the service is alive.
     *
     * @return 'pong'
     */
    @Override
    public String ping() {
        return "pong";
    }

    /**
     * Get the device info.
     *
     * @return device info.
     */
    @Override
    public DeviceInfo deviceInfo() {
        return DeviceInfo.getDeviceInfo();
    }
}

```

```
...
}
```

上面是最基本的几个函数：

- ping
  - 返回 pong
    - 表示服务还在，有效、alive
- deviceInfo
  - 对应着之前调试：
    - d = u2.connect('8c8a4d4d')
  - 期间输出的：
    - conn=<urllib3.connection.HTTPConnection object at  
0x1077f4be0>,method=POST,url=/jsonrpc/0,timeout\_obj=Timeout(connect=2, read=2,  
total=None),body={"jsonrpc": "2.0", "id": 1, "method": "deviceInfo"},headers=  
'User-Agent': 'python-requests/2.22.0', 'Accept-Encoding': 'gzip, deflate',  
'Accept': '\*/\*', 'Connection': 'keep-alive', 'Content-Length': '51'},chunked=False
  - 中的
    - "method": "deviceInfo"
  - 用于返回设备信息

而AutomatorServiceImpl中海油其他很多很多功能的具体实现。下面分别介绍一下之前接触过的。

```
public boolean click(int x, int y) {  
public boolean drag(int startX, int startY, int endX, int endY, int steps) throws NotImplementedException {  
public boolean swipe(int startX, int startY, int endX, int endY, int steps) {  
...  
}
```

都是常见的基础功能。

```
// Multi touch is a little complicated  
@Override  
public boolean injectInputEvent(int action, float x, float y, int metaState) {  
    MotionEvent e = MotionEvent.obtain(SystemClock.uptimeMillis(),  
        SystemClock.uptimeMillis(),  
        action, x, y, metaState);  
    e.setSource(InputDevice.SOURCE_TOUCHSCREEN);  
    boolean b = uiAutomation.injectInputEvent(e, true);  
    e.recycle();  
    return b;  
}
```

之前就遇到过多次，上层调用一些函数会报错，其中就会提到这个

比如：

【已解决】python的uiautomator2报错： uiautomator2.exceptions.JsonRpcError -32601 Method not found data injectInputEvent

中的

```
obj.jsonrpc.injectInputEvent(ACTION_DOWN, x, y, 0)
```

其他还有很多很多：

```
/**  
 * Simulates a short press using key name.  
 *  
 * @param key possible key name is home, back, left, right, up, down, center, menu, se  
arch, enter, delete(or del), recent(recent apps), volume_up, volume_down, volume_mute, cam  
era, power  
 * @return true if successful, else return false  
 * @throws RemoteException  
 */  
@Override  
public boolean pressKey(String key) throws RemoteException {  
    boolean result;  
    key = key.toLowerCase();  
    if ("home".equals(key)) result = device.pressHome();  
    else if ("back".equals(key)) result = device.pressBack();  
    else if ("left".equals(key)) result = device.pressDPadLeft();  
    else if ("right".equals(key)) result = device.pressDPadRight();  
    else if ("up".equals(key)) result = device.pressDPadUp();  
    else if ("down".equals(key)) result = device.pressDPadDown();  
    else if ("center".equals(key)) result = device.pressDPadCenter();  
    else if ("menu".equals(key)) result = device.pressMenu();  
    else if ("search".equals(key)) result = device.pressSearch();  
    else if ("enter".equals(key)) result = device.pressEnter();  
    else if ("delete".equals(key) || "del".equals(key)) result = device.pressDelete();  
    else if ("recent".equals(key)) result = device.pressRecentApps();  
    else if ("volume_up".equals(key)) result = device.pressKeyCode(KeyEvent.KEYCODE_VOL  
UME_UP);  
    else if ("volume_down".equals(key))  
        result = device.pressKeyCode(KeyEvent.KEYCODE_VOLUME_DOWN);  
    else if ("volume_mute".equals(key))  
        result = device.pressKeyCode(KeyEvent.KEYCODE_VOLUME_MUTE);  
    else if ("camera".equals(key)) result = device.pressKeyCode(KeyEvent.KEYCODE_CAMERA  
);  
    else result = "power".equals(key) && device.pressKeyCode(KeyEvent.KEYCODE_POWER);  
  
    return result;  
}  
  
public boolean pressKeyCode(int keyCode) {  
public boolean pressKeyCode(int keyCode, int metaState) {
```

```

public void clearTextField(Selector obj) throws UiObjectNotFoundException {
}

/**
 * Reads the text property of the UI element
 *
 * @param obj the selector of the UiObject.
 * @return text value of the current node represented by this UiObject
 * @throws UiObjectNotFoundException
 */
@Override
public String getText(Selector obj) throws UiObjectNotFoundException {
    if (obj.toUiObject2() == null) {
        return device.findObject(obj.toUiSelector()).getText();
    } else {
        return obj.toUiObject2().getText();
    }
}

/**
 * Sets the text in an editable field, after clearing the field's content. The UiSelector
 * selector of this object must reference a UI element that is editable. When you call this
 * method, the method first simulates a click() on editable field to set focus. The method
 * then clears the field's contents and injects your specified text into the field. If you want
 * to capture the original contents of the field, call getText() first. You can then modify
 * the text and use this method to update the field.
 *
 * @param obj the selector of the UiObject.
 * @param text string to set
 * @return true if operation is successful
 * @throws UiObjectNotFoundException
 */
@Override
public boolean setText(Selector obj, String text) throws UiObjectNotFoundException {
    try {
        obj.toUiObject2().click();
        obj.toUiObject2().setText(text);
        return true;
    } catch (NullPointerException | StaleObjectException e) {
        return device.findObject(obj.toUiSelector()).setText(text);
    }
}

/**
 * Performs a click at the center of the visible bounds of the UI element represented
 * by this UiObject.
 *
 * @param obj the target ui object.

```

```

    * @return true if successful else false
    * @throws UiObjectNotFoundException
    */
@Override
public boolean click(Selector obj) throws UiObjectNotFoundException {
    if (obj.toUiObject2() == null) {
        return device.findObject(obj.toUiSelector()).click();
    } else {
        obj.toUiObject2().click();
        return true;
    }
}

/**
 * Clicks the bottom and right corner or top and left corner of the UI element
 *
 * @param obj    the target ui object.
 * @param corner "br"/"bottomright" means BottomRight, "tl"/"topleft" means TopLeft, "center" means Center.
 * @return true on success
 * @throws UiObjectNotFoundException
 */
@Override
public boolean click(Selector obj, String corner) throws UiObjectNotFoundException {
    return click(device.findObject(obj.toUiSelector()), corner);
}

private boolean click(UiObject obj, String corner) throws UiObjectNotFoundException {
    if (corner == null) corner = "center";
    corner = corner.toLowerCase();
    if ("br".equals(corner) || "bottomright".equals(corner)) return obj.clickBottomRight();
    else if ("tl".equals(corner) || "topleft".equals(corner)) return obj.clickTopLeft();
    else if ("c".equals(corner) || "center".equals(corner)) return obj.click();
    return false;
}

public boolean dragTo(Selector obj, Selector destObj, int steps) throws UiObjectNotFoundException,
    NotImplementedException {
}

/**
 * Performs the swipe up/down/left/right action on the UiObject
 *
 * @param obj    the target ui object.
 * @param dir    "u"/"up", "d"/"down", "l"/"left", "r"/"right"
 * @param steps indicates the number of injected move steps into the system. Steps are

```

```

injected about 5ms apart. So a 100 steps may take about 1/2 second to complete.
 * @return true if successful
 * @throws UiObjectNotFoundException
 */
@Override
public boolean swipe(Selector obj, String dir, int steps) throws UiObjectNotFoundException {
    return swipe(device.findObject(obj.toUiSelector()), dir, steps);
}

private boolean swipe(UiObject item, String dir, int steps) throws UiObjectNotFoundException {
    dir = dir.toLowerCase();
    boolean result = false;
    if ("u".equals(dir) || "up".equals(dir)) result = item.swipeUp(steps);
    else if ("d".equals(dir) || "down".equals(dir)) result = item.swipeDown(steps);
    else if ("l".equals(dir) || "left".equals(dir)) result = item.swipeLeft(steps);
    else if ("r".equals(dir) || "right".equals(dir)) result = item.swipeRight(steps);
    return result;
}

...
...
...

```

其他更多函数就不贴代码了。

## 底层调用dumpWindowHierarchy, 处理, 返回数据

如上所述, AutomatorServiceImpl.java 中的很多功能函数, 此处最关心的dumpWindowHierarchy了:

```

/**
 * Helper method used for debugging to dump the current window's layout hierarchy. The
file root location is /data/local/tmp
 *
 * @param compressed use compressed layout hierarchy or not using setCompressedLayoutH
ierarchy method. Ignore the parameter in case the API level lt 18.
 * @param filename the filename to be stored. @deprecated
 * @return the absolute path name of dumped file.
 */
@Deprecated
@Override
public String dumpWindowHierarchy(boolean compressed, String filename) {
    return dumpWindowHierarchy(compressed);
}

/**
 * Helper method used for debugging to dump the current window's layout hierarchy.
 *
 * @param compressed use compressed layout hierarchy or not using setCompressedLayoutH

```

```

eierarchy method. Ignore the parameter in case the API level lt 18.
    * @return the absolute path name of dumped file.
    */
@Override
public String dumpWindowHierarchy(boolean compressed) {
    device.setCompressedLayoutHeirarchy(compressed);
    ByteArrayOutputStream os = null;
    try {
        os = new ByteArrayOutputStream();
        AccessibilityNodeInfoDumper.dumpWindowHierarchy(device, os);
        device.dumpWindowHierarchy(os);

        return os.toString("UTF-8");
    } catch (IOException e) {
        Log.d("dump Window Hierarchy got IOException " + e);
    }finally {
        if (os != null) {
            try {
                os.close();
            } catch (IOException e) {
                //ignore
            }
        }
    }
}

return null;
}

```

前一个：

```
public String dumpWindowHierarchy(boolean compressed, String filename) {
```

已废弃。

后一个，核心是调用：

```
AccessibilityNodeInfoDumper.dumpWindowHierarchy(device, os);
```

app/src/androidTest/java/com/github/uiautomator/stub/AccessibilityNodeInfoDumper.java

```

public static void dumpWindowHierarchy(UIDevice device, OutputStream out) throws IOException {
    XmlSerializer serializer = Xml.newSerializer();
    serializer.setFeature("http://xmlpull.org/v1/doc/features.html#indent-output", true);
    serializer.setOutput(out, "UTF-8");
    serializer.startDocument("UTF-8", true);
    serializer.startTag("", "hierarchy");
    serializer.attribute("", "rotation", Integer.toString(device.getDisplayRotation()));
}

```

```

    ;
    AccessibilityNodeInfo[] arr$ = getWindowRoots(device); // device.getWindowRoots();
    int len$ = arr$.length;

    for (int i$ = 0; i$ < len$; ++i$) {
        AccessibilityNodeInfo root = arr$[i$];
        dumpNodeRec(root, serializer, 0, device.getDisplayWidth(), device.getDisplayHeight());
    }

    serializer.endTag("", "hierarchy");
    serializer.endDocument();
}

```

最终返回的内容，就是此处的dumpWindowHierarchy函数的处理，生成xml内容后，所返回的。

比如某次调试过程：

jsonrpc的调用：

```
[191120 10:17:07][__init__.py 493] jsonrpc_call: jsonrpc_url http://127.0.0.1:64445/jsonrpc/0, method dumpWindowHierarchy, params (True, None), http_timeout 60
```

底层发送的请求是：

```
Shell$ curl -X POST -d 'b'{"jsonrpc": "2.0", "id": "5a175f3159cc1aa2e27f1cb68f5a0509", "method": "dumpWindowHierarchy", "params": [true, null]}' http://127.0.0.1:64445/jsonrpc/0
```

最终返回的结果是：

```
Output {"jsonrpc":"2.0","id":"5a175f3159cc1aa2e27f1cb68f5a0509","result":"<?xml version='1.0' encoding='UTF-8' standalone='yes' ?><hierarchy rotation=\"0\"><node index=\"0\" text=\"\" resource-id=\"\" class=\"android.widget.FrameLayout\" package=\"com.tencent.mm\" content-desc=\"当前所在页面,搜一搜\" checkable=\"false\" checked=\"false\" clickable=\"false\" enabled=\"true\" focusable=\"false\" focused=\"false\" scrollable=\"false\" long-clickable=\"false\" password=\"false\" selected=\"false\" bounds=\"[0,0][720,1208]\"><node index=\"0\" text=\"\" resource-id=\"com.tencent.mm:id/om\" class=\"android.widget.FrameLayout\" package=\"com.tencent.mm\" content-desc=\"containerFrameLayout\" checkable=\"false\" checked=\"false\" clickable=\"false\" enabled=\"true\" focusable=\"false\" focused=\"false\" scrollable=\"false\" long-clickable=\"false\" password=\"false\" selected=\"false\" bounds=\"[0,134][720,1208]\"><node index=\"0\" text=\"\" resource-id=\"com.tencent.mm:id/boe\" class=\"android.widget.FrameLayout\" package=\"com.tencent.mm\" content-desc=\"containerFrameLayout\" checkable=\"false\" checked=\"false\" clickable=\"false\" enabled=\"true\" focusable=\"false\" focused=\"false\" scrollable=\"false\" long-clickable=\"false\" password=\"false\" selected=\"false\" bounds=\"[0,134][720,1208]\"><node NAF=\"true\" index=\"0\" text=\"\" resource-id=\"\" class=\"com.tencent.tbs.core.webkit.WebView\" package=\"com.tencent.mm\" content-desc=\"\" checkable=\"false\" checked=\"false\" clickable=\"true\" enabled=\"true\" focusable=\"true\" focused=\"true\" scrollable=\"false\" long-clickable=\"true\" password=\"false\" selected=\"false\" bounds=\"[0,134][720,1208]\">
```

```

word=false" selected=false" bounds="[0,134][720,1208]" /></node></node index
= "1" text="" resource-id="com.tencent.mm:id/m4" class="android.widget.LinearLayout"
" package="com.tencent.mm" content-desc="" checkable=false" checked=false" click
able=true" enabled=true" focusable=false" focused=false" scrollable=false"
long-clickable=false" password=false" selected=false" bounds="[0,48][71,134]" ><
node index="0" text="" resource-id="com.tencent.mm:id/m5" class="android.widget.Im
ageView" package="com.tencent.mm" content-desc="返回" checkable=false" checked=false"
clickable=false" enabled=true" focusable=false" focused=false" scrollable=false"
long-clickable=false" password=false" selected=false" bounds="[0,48][71,134]" ></node><node index="3" text="gh_cfcfcee032cc" resource-id="com.tencent.mm
:id/m7" class="android.widget.EditText" package="com.tencent.mm" content-desc=""
checkable=false" checked=false" clickable=true" enabled=true" focusable=true"
focused=false" scrollable=false" long-clickable=true" password=false" selecte
d=false" bounds="[116,48][706,134]" ><node NAF=true" index="4" text="" resour
ce-id="com.tencent.mm:id/m3" class="android.widget.ImageButton" package="com.tencent.
mm" content-desc="" checkable=false" checked=false" clickable=true" enabled=true"
focusable=true" focused=false" scrollable=false" long-clickable=false"
password=false" selected=false" bounds="[663,69][706,112]" ></node></hierarchy">

```

可见其中的xml头部的内容：

```
<?xml version='1.0' encoding='UTF-8' standalone='yes' ?><hierarchy rotation="0">
```

就是上面的 `XmlSerializer` 的代码所生成的。

而其他的node节点，则是`dumpNodeRec`所生成的。

由此，后续深入研究，才知道，最终返回的节点中，如果符合NAF条件，则会被忽略其下内容，最终返回一个NAF="true"的节点，导致后续只返回部分页面内容的。

具体细节详见：

- 【未解决】 uiautomator2中`dump_hierarchy`中只能获取到页面的部分的xml源码
- 【已解决】 搞懂uiautomator-server中最终的底层实现`dumpWindowHierarchy`的处理返回页面数据的逻辑

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2020-08-09 10:14:28

## 常用代码段

开发uiautomator2期间，把一些常用的功能，常用代码段，封装成了通用的函数并贴出来，和具体调用方式，供参考。

其中后续各种通用功能和函数，往往都会调用到一些基础的工具类函数，详见接下来的工具类函数，后续就不再赘述。

crifan.com，使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2021-07-18 09:59:23

# 工具类函数

在介绍通用功能之前，要先把常用到的基础的工具类的函数贴出来，供参考使用。

## 获取命令执行后返回的结果

```
def get_cmd_lines(cmd, text=False):
    # 执行cmd命令，将结果保存为列表
    resultStr = ""
    resultStrList = []
    try:
        consoleOutputByte = subprocess.check_output(cmd, shell=True) # b'C02Y3N10JHC8\n'
        try:
            resultStr = consoleOutputByte.decode("utf-8")
        except UnicodeDecodeError:
            # TODO: use chardet auto detect encoding
            # consoleOutputStr = consoleOutputByte.decode("gbk")
            resultStr = consoleOutputByte.decode("gb18030")

        if not text:
            resultStrList = resultStr.splitlines()
    except Exception as err:
        print("err=%s when run cmd=%s" % (err, cmd))

    if text:
        return resultStr
    else:
        return resultStrList
```

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09 10:14:28

## adb

uiautomator2操作安卓设备期间，往往会涉及到，借助于安卓体系内本身就有的工具 `adb`，去实现对设备的一些操控。

此处整理出一些常见的用法和通用功能。

### 获取当前安卓手机名

```
def get_phone_name_Android(self):
    # cmd = 'adb -s {} shell getprop ro.product.model'.format(self.device)
    cmd = 'adb -s {} shell getprop ro.product.name'.format(self.device)
    text = CommonUtils.get_cmd_lines(cmd, text=True)
    # https://miuiver.com/xiaomi-device-codename/
    #     begonia -> 红米Note 8内部代号为 "begonia"
    return re.sub("\s", "", text)
    # isRunCmdOk, outputText = self.getCommandOutput(cmd)
    # if isRunCmdOk:
    #     phoneName = outputText
    # else:
    #     phoneName = ""
    # return phoneName
```

调用：

```
def get_phone_name(self):
    # 获取手机名称，以提取配置信息
    if self.isAndroid:
        return self.get_phone_name_Android()
```

### 获取当前连接的设备

```
def get_devices_Android(self):
    lines = CommonUtils.get_cmd_lines('adb devices')
    return [line.split()[0] for line in lines if line and not "devices" in line]
```

调用：

```
devices = self.get_devices_Android()
```

相关命令输出举例：

```
→ ~ adb devices
List of devices attached
```

```
8c8a4d4d      device
```

## 优化版：获取安卓设备列表

```
def getAndroidDeviceList(self, isGetDetail=False):
    """Get android device list

    Args:
        isGetDetail (bool): True to use `adb devices -l`, False to use `adb devices`

    Returns:
        device list(list)
    Raises:
    Examples:
        output:
            False -> ['2e2a0cb1', 'orga4pmzee4ts47t', '192.168.31.84:5555']
            True -> [{('2e2a0cb1': {'usb': '338952192X', 'product': 'PD2065', 'model': 'V2065A', 'device': 'PD2065', 'transport_id': '4'}), {'orga4pmzee4ts47t': {'usb': '338886656X', 'product': 'atom', 'model': 'M2004J7AC', 'device': 'atom', 'transport_id': '24'}}, {'192.168.31.84:5555': {'product': 'PD2065', 'model': 'V2065A', 'device': 'PD2065', 'transport_id': '5'}}]
    """
    deviceList = []

    getDevicesCmd = 'adb devices'
    if isGetDetail:
        getDevicesCmd += " -l"
    logging.debug("getDevicesCmd=%s", getDevicesCmd)

    deviceLines = CommonUtils.get_cmd_lines(getDevicesCmd)
    logging.debug("deviceLines=%s", deviceLines)
    # ['List of devices attached', '2e2a0cb1\tdevice', 'orga4pmzee4ts47t\tdevice', '192.168.31.84:5555\tdevice', '']

    """
    adb devices :
        List of devices attached
        2e2a0cb1      device
        orga4pmzee4ts47t      device
        192.168.31.84:5555      device
    """

    """
    adb devices -l:
        List of devices attached
        2e2a0cb1              device usb:338952192X product:PD2065 model:V2065A device:PD2065 transport_id:4
        orga4pmzee4ts47t      device usb:338886656X product:atom model:M2004J7AC device:atom transport_id:24
        192.168.31.84:5555      device product:PD2065 model:V2065A device:PD2065 transport_id:5
    """

    for eachLine in deviceLines:
```

```

if not eachLine:
    continue

if "devices attached" in eachLine:
    continue

foundDevice = re.search("(?P<devSerial>[\w.\.:]+)\s+device\s*(?P<devDetail>[\w\.: ]+)?", eachLine)
    logging.debug("foundDevice=%s", foundDevice)
    # foundDevice=<re.Match object; span=(0, 101), match='2e2a0cb1'>
    # devSerial=usb:338952192X prod>
    if foundDevice:
        devSerial = foundDevice.group("devSerial")
        logging.debug("devSerial=%s", devSerial)
        # devSerial=2e2a0cb1
        if isGetDetail:
            devDetail = foundDevice.group("devDetail")
            logging.debug("devDetail=%s", devDetail)
            # devDetail=usb:338952192X product:PD2065 model:V2065A device:PD2065 transport_id:4
            keyValueIter = re.finditer("(?P<key>\w+):(?P<value>\w+)", devDetail) # <callable_iterator object at 0x10baa3a60>
            keyValueMatchList = list(keyValueIter)
            logging.debug("keyValueMatchList=%s", keyValueMatchList)
            # keyValueMatchList=[<re.Match object; span=(0, 14), match='usb:338952192X'>, <re.Match object; span=(15, 29), match='product:PD2065'>, <re.Match object; span=(30, 42), match='model:V2065A'>, <re.Match object; span=(43, 56), match='device:PD2065'>, <re.Match object; span=(57, 71), match='transport_id:4'>]
            detailInfoDict = {}
            for eachMatch in keyValueMatchList:
                eachKey = eachMatch.group("key")
                eachValue = eachMatch.group("value")
                detailInfoDict[eachKey] = eachValue
            logging.debug("detailInfoDict=%s", detailInfoDict)
            # detailInfoDict={'usb': '338952192X', 'product': 'PD2065', 'model': 'V2065A', 'device': 'PD2065', 'transport_id': '4'}
            curDevDetailDict = {
                devSerial: detailInfoDict
            }
            logging.debug("curDevDetailDict=%s", curDevDetailDict)
            # curDevDetailDict={'2e2a0cb1': {'usb': '338952192X', 'product': 'PD2065', 'model': 'V2065A', 'device': 'PD2065', 'transport_id': '4'}}
            deviceList.append(curDevDetailDict)
        else:
            deviceList.append(devSerial)

logging.debug("deviceList=%s", deviceList)
# deviceList=[{'2e2a0cb1': {'usb': '338952192X', 'product': 'PD2065', 'model': 'V2065A', 'device': 'PD2065', 'transport_id': '4'}}, {'orga4pmzee4ts47t': {'usb': '338886656X', 'product': 'atom', 'model': 'M2004J7AC', 'device': 'atom', 'transport_id': '24'}}, {'192.168.31.84:5555': {'product': 'PD2065', 'model': 'V2065A', 'device': 'PD2065', 'transport_id': '5'}}]
# ['2e2a0cb1', 'orga4pmzee4ts47t', '192.168.31.84:5555']

```

```
    return deviceList
```

调用：

```
deviceDetailList = self.getAndroidDeviceList(isGetDetail=False)
# ['2e2a0cb1', 'orga4pmzee4ts47t', '192.168.31.84:5555']
```

或：

```
deviceDetailList = self.getAndroidDeviceList(isGetDetail=True)
# [{'2e2a0cb1': {'usb': '338952192X', 'product': 'PD2065', 'model': 'V2065A', 'device': 'PD2065', 'transport_id': '4'}}, {'orga4pmzee4ts47t': {'usb': '338886656X', 'product': 'atom', 'model': 'M2004J7AC', 'device': 'atom', 'transport_id': '24'}}, {'192.168.31.84:5555': {'product': 'PD2065', 'model': 'V2065A', 'device': 'PD2065', 'transport_id': '5'}}]
```

## 检测安卓设备是否连接

```
def isAndroidUsbConnected(self, deviceSerialId):
    """Check whether android device is currently USB wired connected or not

    Args:
        deviceSerialId (str): android devivce serial id
    Returns:
        connected or not (bool)
    Raises:
    Examples:
        input: "orga4pmzee4ts47t"
        output: True
    """
    isUsbConnected = False
    isRealSerialId = re.search("\w+", deviceSerialId)
    if not isRealSerialId:
        # makesure is not wifi, such as: 192.168.31.84:5555
        logging.error("Invalid android USB wired connected device serial id %s", deviceSerialId)
    return isUsbConnected

    deviceDetailList = self.getAndroidDeviceList(isGetDetail=True)
    for eachDevDetailDict in deviceDetailList:
        curDevSerialStr, curDevDetailDict = list(eachDevDetailDict.items())[0]
        if deviceSerialId == curDevSerialStr:
            detailInfoKeyList = list(curDevDetailDict.keys())
            # ['usb', 'product', 'model', 'device', 'transport_id']
            if "usb" in detailInfoKeyList:
                isUsbConnected = True
                break

    return isUsbConnected
```

调用：

```
deviceId = "orga4pmzee4ts47t"
isUsbConnected = self.isAndroidUsbConnected(deviceId)
```

## 用adb通过WiFi连接设备

```
def androidConnectWiFiDevice(self, wifiSerial):
    """Use Android `adb connect` to connect WiFi wireless devive

    Args:
        wifiSerial (str): android devivce WiFi serial, eg: 192.168.31.84:5555
    Returns:
        connect ok or not (bool)
    Raises:
    Examples:
        input: "192.168.31.84:5555"
        output: True
    """
    isConnectOk = False

    adbConnectCmd = "adb connect %s" % wifiSerial
    logging.info("Try connect Android device: %s", adbConnectCmd)
    # os.system(adbConnectCmd) # when failed, will wait too long time: ~ 1 minutes
    cmdOutputStr = CommonUtils.get_cmd_lines(adbConnectCmd, text=True, timeout=1)
    logging.info("console output: %s", cmdOutputStr)
    # connected to 192.168.31.84:5555
    # already connected to 192.168.31.84:5555
    # failed to connect to '192.168.31.84:5555': Operation timed out
    # "failed to connect to '192.168.31.84:5555': Connection refused\n"
    # err=Command 'adb connect 192.168.31.84:5555' timed out after 1 seconds when run cmd=
    adb connect 192.168.31.84:5555
    if cmdOutputStr:
        if "connected" in cmdOutputStr:
            isConnectOk = True
        elif ("failed" in cmdOutputStr) or ("timed out" in cmdOutputStr):
            isConnectOk = False
    else:
        isConnectOk = False

    return isConnectOk
```

调用：

```
devWifiSerialId = "192.168.31.84:5555"
isWiFiConnected = self.androidConnectWiFiDevice(devWifiSerialId)
```

## 获取当前正在运行的app和页面activity

```

def get_PackageActivity_Android(self):
    # adb直接获取当前活跃app及activity
    package, activity = "", ""
    cmd = ['dumpsys activity |grep {}'.format(item) for item in ['mFocusedActivity', 'mResumedActivity']]
    for cmd in cmd:
        output = self.driver.shell(cmd).output
        result = re.search("u0(.*)/", output)
        package = result.group(1).strip() if result else ""
        result = re.search("/(.*)\s", output)
        activity = result.group(1).strip() if result else ""
        if package and activity:
            return package, activity
    return package, activity

```

调用：

```
package, activity = self.get_PackageActivity()
```

## 获取已安装app列表

```

def get_packages(self):
    # 获取已安装的app的appPackage列表
    if isinstance(self.driver, u2.UIAutomatorServer):
        text = self.driver.shell("pm list packages")[0]
        return re.findall(':(.*?)\n', text)
    else:
        cmd = 'adb -s {0} shell pm list packages'.format(self.device)
        lines = CommonUtils.get_cmd_lines(cmd)
        return {line.split(":")[1].strip() for line in lines}

```

调用：

```
packages = self.get_packages()
```

## 安装安卓app

```

def install_app_Android(self, item, packages=None):
    if packages is None:
        packages = self.get_packages()
    if item[1] in packages:
        logging.info("AppName {0} is already installed".format(item[0]))
    else:
        logging.info("start to install app in {}".format(os.path.basename(self.arg_options.task)))
        os.system("adb -s {0} install {1}".format(self.device, item[3]))

```



调用：

```
def install_app(self, item, packages=None):
    # 安装app
    if self.isAndroid:
        return self.install_app_Android(item, packages)
```

## 卸载安卓app

```
def uninstall_app(self, item):
    # 卸载安装包
    os.system("adb -s {} uninstall {}".format(self.device, item[1]))
    logging.info("uninstall app {} end".format(item[1]))
```

调用：

```
if item[1] in packages:
    self.uninstall_app(item)
```

## 判断屏幕是否已解锁

```
def is_device_unlock_Android(self, device):
    os.system('adb -s {} shell input keyevent 3'.format(device))
    time.sleep(1)
    # cmd = [
    #     'adb -s {} shell dumpsys window policy | grep isStatusBarKeyguard'.format(device),
    # ],
    #     'adb -s {} shell dumpsys window policy | grep mShowingLockscreen'.format(device),
    #     'adb -s {} shell dumpsys window policy | grep mDreamingLockscreen'.format(device),
    # ],
    # ]
    # for cmd in cmd:
    #     text = CommonUtils.get_cmd_lines(cmd, text=True)
    #     if text and "=true" in text:
    #         logging.info("start to unlock device {}".format(device))
    #         return False

    # cmd = [
    #     'adb -s {} shell dumpsys power | grep mHoldingDisplaySuspendBlocker'.format(device),
    # ],
    # ]
    # for cmd in cmd:
    #     text = CommonUtils.get_cmd_lines(cmd, text=True)
    #     if text and "=false" in text:
    #         # ' mHoldingDisplaySuspendBlocker=false\n'
```

```

#         logging.info("start to unlock device {}".format(device))
#         return False

checkCmds = 'adb -s {} shell dumpsys window | grep mDreamingLockscreen'.format(device)
text = CommonUtils.get_cmd_lines(checkCmds, text=True)
if text and "mDreamingLockscreen=true" in text:
    # '    mShowingDream=false mDreamingLockscreen=true mDreamingSleepToken=null\n'
    logging.info("start to unlock device {}".format(device))
    return False

logging.info("device {} is already unlock".format(device))
return True

```

调用：

```

if self.isAndroid:
    return self.is_device_unlock_Android(device)

```

## 获取安卓手机电量

```

def get_device_electricity_Android(self):
    shell_cmd = 'dumpsys battery | grep level'
    adb_cmd = 'adb -s {0} shell {1}'.format(self.device.shell_cmd)
    # level = self.driver.shell(shell_cmd).output if isinstance(self.driver, u2.UIAutomatorServer) else CommonUtils.get_cmd_lines(adb_cmd, text=True)
    level = self.driver.shell(shell_cmd).output if isinstance(self.driver, u2.Device) else CommonUtils.get_cmd_lines(adb_cmd, text=True)
    result = re.search("\d+", level)
    ratio = int(result.group()) if result else 100
    return ratio

```

调用：

```

batteryElectricityPercentInt = self.get_device_electricity_Android()

```

## 获取安卓手机名

```

def get_phone_name_Android(self):
    # cmd = 'adb -s {} shell getprop ro.product.model'.format(self.device)
    cmd = 'adb -s {} shell getprop ro.product.name'.format(self.device)
    text = CommonUtils.get_cmd_lines(cmd, text=True)
    # https://miuiver.com/xiaomi-device-codename/
    #     begonia -> 红米Note 8内部代号为“begonia”
    return re.sub("\s", "", text)
    # isRunCmdOk, outputText = self.getCommandOutput(cmd)
    # if isRunCmdOk:

```

```
#     phoneName = outputText
# else:
#     phoneName = ""
# return phoneName
```

调用：

```
if self.isAndroid:
    return self.get_phone_name_Android()
```

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2021-07-18 09:59:23

## 设备相关

此处整理出，和安卓设备相关的一些通用功能的函数和调用举例。

### 获取安卓设备信息

```
def getDeviceInfo(self):
    return self.driver.device_info
```

调用：

```
deviceInfo = self.getDeviceInfo()
logging.info("deviceInfo=%s" % deviceInfo)
```

输出举例：

```
# deviceInfo={'udid': '2e2a0cb1-36:59:fa:77:bb:a6-V2065A', 'version': '10', 'serial': '2e2a0cb1', 'brand': 'vivo', 'model': 'V2065A', 'hwaddr': '36:59:fa:77:bb:a6', 'port': 7912, 'sdk': 29, 'agentVersion': '0.9.5', 'display': {'width': 720, 'height': 1600}, 'battery': {'acPowered': True, 'usbPowered': False, 'wirelessPowered': False, 'status': 5, 'health': 2, 'present': True, 'level': 100, 'scale': 100, 'voltage': 4406, 'temperature': 286, 'technology': 'Li-poly'}, 'memory': {'total': 5832196, 'around': '6 GB'}, 'cpu': {'cores': 8, 'hardware': 'Qualcomm Technologies, Inc BENGAL'}, 'arch': '', 'owner': None, 'presenceChangeAt': '0001-01-01T00:00:00Z', 'usingBeganAt': '0001-01-01T00:00:00Z', 'product': None, 'provider': None}
```

### 获取(u2)驱动信息

代码：

```
driverInfo = self.driver.info
logging.info("driverInfo=%s" % driverInfo)
```

输出举例：

```
# driverInfo={'currentPackageName': 'com.bbk.launcher2', 'displayHeight': 1459, 'displayRotation': 0, 'displaySizeDpX': 360, 'displaySizeDpY': 800, 'displayWidth': 720, 'productName': 'PD2065', 'screenOn': True, 'sdkInt': 29, 'naturalOrientation': True}
```

### 获取安卓版本

```
def getAndroidVersion(self):
```

```
"""返回安卓版本号, float值: 6.0, 9.0 """
deviceInfo = self.getDeviceInfo()
logging.debug("deviceInfo=%s" % deviceInfo)
androidVersionStr = deviceInfo["version"] # '6.0'
androidVersionFloat = float(androidVersionStr)
return androidVersionFloat
```

调用:

```
curAndroidVersionFloat = self.getAndroidVersion()
ANDROID_VERSION_NEED_RESTART_U2 = 7.0
if curAndroidVersionFloat <= ANDROID_VERSION_NEED_RESTART_U2:
    isNeedRestartU2 = True
```

## 获取安卓屏幕分辨率

```
def getCurScreenResolution(self):
    """Get current screen resolution"""
    driverInfo = self.driver.info
    logging.debug("driverInfo=%s" % driverInfo)
    # displayWidth = driverInfo["displayWidth"]
    # displayHeight = driverInfo["displayHeight"]
    # logging.info("displayWidth=%s, displayHeight=%s", displayWidth, displayHeight)
    # deviceInfo = self.driver.device_info
    deviceInfo = self.getDeviceInfo()
    logging.debug("deviceInfo=%s" % deviceInfo)
    deviceDisplay = deviceInfo["display"]
    logging.debug("deviceDisplay=%s" % deviceDisplay)
    screenWidth = deviceDisplay["width"]
    screenHeight = deviceDisplay["height"]
    logging.debug("screenWidth=%s, screenHeight=%s", screenWidth, screenHeight)
    if driverInfo["displayRotation"]:
        curScreenWidth = screenHeight
        curScreenHeight = screenWidth
    else:
        curScreenWidth = screenWidth
        curScreenHeight = screenHeight
    logging.debug("curScreenWidth=%s, curScreenHeight=%s", curScreenWidth, curScreenHeight)

    return (curScreenWidth, curScreenHeight)
```

调用:

```
screenWidth, screenHeight = self.getCurScreenResolution()
```

输出:

```
[191213 16:16:13][AppCrawler.py 209] driverInfo {'currentPackageName': 'com.zztl.dragon.cs',  
'displayHeight': 720, 'displayRotation': 1, 'displaySizeDpX': 640, 'displaySizeDpY': 360,  
'displayWidth': 1196, 'productName': 'DIG-AL00', 'sdkInt': 23, 'naturalOrientation': False}  
[191213 16:16:13][AppCrawler.py 212] displayWidth 1196, displayHeight 720  
[191213 16:16:13][AppCrawler.py 214] deviceInfo {'udid': 'DWH9X17124W03779-18:d2:76:f4:9e:7b-DIG-AL00',  
'version': '6.0', 'serial': 'DWH9X17124W03779', 'brand': 'HUAWEI', 'model':  
'DIG-AL00', 'hwaddr': '18:d2:76:f4:9e:7b', 'port': 7912, 'sdk': 23, 'agentVersion': '0.7.4'  
, 'display': {'width': 720, 'height': 1280}, 'battery': {'acPowered': True, 'usbPowered':  
False, 'wirelessPowered': False, 'status': 2, 'health': 2, 'present': True, 'level': 26,  
'scale': 100, 'voltage': 3892, 'temperature': 315, 'technology': 'Li-ion'}, 'memory': {'total':  
2879816, 'around': '3 GB'}, 'cpu': {'cores': 8, 'hardware': 'Qualcomm Technologies, Inc  
MSM8940'}, 'arch': '', 'owner': None, 'presenceChangedAt': '0001-01-01T00:00:00Z', 'usingBeganAt':  
'0001-01-01T00:00:00Z', 'product': None, 'provider': None}  
[191213 16:16:13][AppCrawler.py 216] deviceDisplay {'width': 720, 'height': 1280}  
[191213 16:16:13][AppCrawler.py 219] screenWidth 720, screenHeight 1280  
[191213 16:16:13][AppCrawler.py 226] curScreenWidth=1280, curScreenHeight=720
```

得到了我们要的：屏幕的宽度和高度

且知道了是当前屏幕是否已旋转（从安卓手机的默认的竖屏，旋转成游戏的横屏）了

另外，当屏幕故意不去旋转，回到默认竖屏后：



此时

- 旋转为 False
  - displayRotation : 0
  - naturalOrientation : True
- 但 displayHeight 值有变化：是 1208
  - 却不是 1280

如图：

```

197     # payYuanStr = "%d元"
198     # payYuanImgPath = "debug/安卓/app/Gam
199     # foundPayYuanStr, foundResultList,
200
201     # # for debug
202     # # jianLingLongImgPath = "debug/安卓
203     # self._isAnnouncementPage(jianLing
204     # zhizhunTuLongImgPath = "debug/安卓
205     # self._isAnnouncementPage(zhizhunTu
206
207
208     driverInfo = self.driver.info
209     logging.info("driverInfo=%s" % driverInfo)
210     displayWidth = driverInfo["displayWidth"]
211     displayHeight = driverInfo["displayHeight"]
212     logging.info("displayWidth=%s, displayHeight=%s", displayWidth, displayHeight)
213     deviceInfo = self.driver.device_info
214     logging.info("deviceInfo=%s" % deviceInfo)
215     deviceDisplay = deviceInfo["display"]
216     logging.info("deviceDisplay=%s" % deviceDisplay)
217     screenWidth = deviceDisplay["width"]
218     screenHeight = deviceDisplay["height"]
219     logging.info("screenWidth=%s, screenHeight=%s", screenWidth, screenHeight)
220     if driverInfo["displayRotation"]:
221         curScreenWidth = screenHeight
222         curScreenHeight = screenWidth
223     else:
224         curScreenWidth = screenWidth
225         curScreenHeight = screenHeight
226     logging.info("curScreenWidth=%s, curScreenHeight=%s", curScreenWidth, curScreenHeight)
227
228     # for debug
229     while self._isAnnouncementPage():

```

问题 438 输出 调试控制台 终端

```

[191213 16:20:47][AppCrawler.py 209] driverInfo: {'currentPackageName': 'com.huawei.android.launcher', 'displayHeight': 1208, 'displayRotation': 0, 'displaySizeDpx': 360, 'displaySizeDpY': 640, 'displayWidth': 720, 'productName': 'DIG-AL00', 'sdkInt': 23, 'naturalOrientation': True}
[191213 16:20:50][AppCrawler.py 212] displayWidth=720, displayHeight=1208
[191213 16:23:17][AppCrawler.py 214] deviceInfo: {'udid': 'DWH9X17124W03779-18:d2:76:f4:9e:7b-DIG-AL00', 'version': '6.0', 'serial': 'DWH9X17124W03779', 'brand': 'HUAWEI', 'model': 'DIG-AL00', 'hwaddr': '18:d2:76:f4:9e:7b', 'port': 7912, 'sdk': 23, 'agentVersion': '0.7.4', 'display': {'width': 720, 'height': 1280}, 'battery': {'acPowered': True, 'usbPowered': False, 'wirelessPowered': False, 'status': 2, 'health': 2, 'present': True, 'level': 26, 'scale': 100, 'voltage': 3892, 'temperature': 315, 'technology': 'Li-ion'}, 'memory': {'total': 2879816, 'around': '3 GB'}, 'cpu': {'cores': 8, 'hardware': 'Qualcomm Technologies, Inc MSM8940'}, 'arch': '', 'owner': None, 'presenceChangedAt': '0001-01-01T00:00:00Z', 'usingBeganAt': '0001-01-01T00:00:00Z', 'product': None}
[191213 16:23:19][AppCrawler.py 216] deviceDisplay={'width': 720, 'height': 1280}
[191213 16:23:21][AppCrawler.py 219] screenWidth=720, screenHeight=1280
[191213 16:23:24][AppCrawler.py 226] curScreenWidth=720, curScreenHeight=1280

```

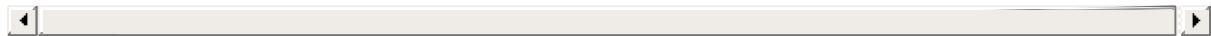
整张图片 1280x720 行 215, 列 46 制表符长度: 4 UTF-8 CRLF Python ⚙ 2

详细log：

```

[191213 16:20:47][AppCrawler.py 209] driverInfo: {'currentPackageName': 'com.huawei.android.launcher', 'displayHeight': 1208, 'displayRotation': 0, 'displaySizeDpx': 360, 'displaySizeDpY': 640, 'displayWidth': 720, 'productName': 'DIG-AL00', 'sdkInt': 23, 'naturalOrientation': True}
[191213 16:20:50][AppCrawler.py 212] displayWidth 720, displayHeight 1208
[191213 16:23:17][AppCrawler.py 214] deviceInfo: {'udid': 'DWH9X17124W03779-18:d2:76:f4:9e:7b-DIG-AL00', 'version': '6.0', 'serial': 'DWH9X17124W03779', 'brand': 'HUAWEI', 'model': 'DIG-AL00', 'hwaddr': '18:d2:76:f4:9e:7b', 'port': 7912, 'sdk': 23, 'agentVersion': '0.7.4', 'display': {'width': 720, 'height': 1280}, 'battery': {'acPowered': True, 'usbPowered': False, 'wirelessPowered': False, 'status': 2, 'health': 2, 'present': True, 'level': 26, 'scale': 100, 'voltage': 3892, 'temperature': 315, 'technology': 'Li-ion'}, 'memory': {'total': 2879816, 'around': '3 GB'}, 'cpu': {'cores': 8, 'hardware': 'Qualcomm Technologies, Inc MSM8940'}, 'arch': '', 'owner': None, 'presenceChangedAt': '0001-01-01T00:00:00Z', 'usingBeganAt': '0001-01-01T00:00:00Z', 'product': None}
[191213 16:23:19][AppCrawler.py 216] deviceDisplay={'width': 720, 'height': 1280}
[191213 16:23:21][AppCrawler.py 219] screenWidth 720, screenHeight 1280
[191213 16:23:24][AppCrawler.py 226] curScreenWidth=720, curScreenHeight=1280

```



详见：

【已解决】uiautomator2获取当前屏幕的宽和高即屏幕大小分辨率信息

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2021-07-18 09:59:23

# 其他

## 获取元素属性

```
# 获取元素属性值
def get_ElementBounds(self, element):
    # 将元素坐标转成数组
    if self.isAndroid:
        # <node index="1" text="" resource-id="com.tencent.mm:id/l1" . . . bounds="[0,48]
[720,134]">
        bounds = element.attrib.get("bounds")
        return list(map(int, re.sub('\[|,|\]', " ", bounds).split())) if bounds else [0, 0,
0, 0]
    elif self.isiOS:
        # <XCUIElementTypeButton . . . name="返回" label="返回" . . . x="20" y="20" width=
"30" height="44"/>
        # attrib = element.attrib
        # xStr = attrib["x"]
        # yStr = attrib["y"]
        # widthStr = attrib["width"]
        # heightStr = attrib["height"]
        # x = int(xStr)
        # y = int(yStr)
        # width = int(widthStr)
        # height = int(heightStr)
        x = self.get_ElementX(element)
        y = self.get_ElementY(element)
        width = self.get_ElementWidth(element)
        height = self.get_ElementHeight(element)
        x1 = x + width
        y1 = y + height
        boundList = [x, y, x1, y1]
        return boundList # [16, 20, 46, 64]

def get_ElementX(self, element):
    if self.isAndroid:
        bounds = self.get_ElementBounds(element)
        x = bounds[0]
        return x
    elif self.isiOS:
        attrib = element.attrib
        xStr = attrib["x"]
        x = int(xStr)
        return x

def get_ElementY(self, element):
    if self.isAndroid:
        bounds = self.get_ElementBounds(element)
        y = bounds[1]
        return y
```

```

    elif self.isiOS:
        attrib = element.attrib
        yStr = attrib["y"]
        y = int(yStr)
        return y

    def get_ElementWidth(self, element):
        if self.isAndroid:
            bounds = self.get_ElementBounds(element)
            width = bounds[2] - bounds[0]
            return width
        elif self.isiOS:
            attrib = element.attrib
            widthStr = attrib["width"]
            width = int(widthStr)
            return width

    def get_ElementHeight(self, element):
        if self.isAndroid:
            bounds = self.get_ElementBounds(element)
            height = bounds[3] - bounds[1]
            return height
        elif self.isiOS:
            attrib = element.attrib
            heightStr = attrib["height"]
            height = int(heightStr)
            return height

    def get_ElementSize(self, element):
        # 获取元素方框大小
        bounds = self.get_ElementBounds(element)
        return (bounds[2] - bounds[0]) * (bounds[3] - bounds[1])

    def get_ElementPoint(self, element):
        # 获取元素中心点坐标
        bounds = self.get_ElementBounds(element)
        return [(bounds[2] + bounds[0])//2, (bounds[3] + bounds[1])//2]

    def get_ElementText(self, element):
        if self.isAndroid:
            # 返回元素text文本
            textKey = "text"
        elif self.isiOS:
            # 返回元素label
            textKey = "label"
        textValue = element.attrib.get(textKey, "")
        return textValue

    def get_ElementContentdesc(self, element):
        if self.isAndroid:
            # 返回元素content-desc文本
            descKey = "content-desc"
        elif self.isiOS:

```

```

    # 返回元素value
    descKey = "value"
descValue = element.attrib.get(descKey, "")
return descValue

def get_ElementDescribe(self, element):
    # # 返回元素text文本和content-desc文本
    # 返回元素 文本 和 描述
    elementText = self.get_ElementText(element)
    elementContentDesc = self.get_ElementContentdesc(element)
    descText = elementText + elementContentDesc
    return descText

```

## 判断是否是布局类型的元素

```

def is_element_layout_Android(self, element):
    # 判断元素是否是out类型(如LinearLayout、RelativeLayout)
    # return "Layout" in element.attrib.get("class")
    curClass = element.attrib.get("class")
    #TODO: 换成re正则匹配 xxxLayout ?
    isLayout = "Layout" in curClass
    # 可能:
    #     android.widget.FrameLayout
    #     android.widget.LinearLayout
    #     android.widget.RelativeLayout
    # for debug
    if isLayout:
        knownLayoutList = [
            "android.widget.FrameLayout",
            "android.widget.LinearLayout",
            "android.widget.RelativeLayout",
        ]
        foundNew = curClass not in knownLayoutList
        if foundNew:
            print("curClass=%s" % curClass)
return isLayout

```

## 判断元素是否是某种类型

```

def is_element_Button(self, element):
    # 元素是否为Button
    if self.isAndroid:
        # return "Button" in element.attrib.get("class")
        return self.is_element_SomeType_Android(element, "Button")
    elif self.isiOS:
        # <XCUIElementTypeButton type="XCUIElementTypeButton" name="返回" label="返回" enabled="true" visible="true" x="20" y="20" width="30" height="44"/>
        # iOSTagButton = "XCUIElementTypeButton"

```

```

# elementTag = element.tag
# isButton = elementTag == iOSTagButton
# return isButton
return self.is_element_SomeType_iOS(element, "XCUIElementTypeButton")

def is_element_Image(self, element):
    # 元素是否为ImageView
    if self.isAndroid:
        # return "Image" in element.attrib.get("class")
        return self.is_element_SomeType_Android(element, "Image")
    elif self.isiOS:
        # <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="false"
        x="0" y="64" width="414" height="56"/>
        # iOSTagImage = "XCUIElementTypeImage"
        # elementTag = element.tag
        # isImage = elementTag == iOSTagImage
        # return isImage
        return self.is_element_SomeType_iOS(element, "XCUIElementTypeImage")

def is_element_EditText(self, element):
    if self.isAndroid:
        # 元素是否为EditText
        # return "EditText" in element.attrib.get("class")
        return self.is_element_SomeType_Android(element, "EditText")
    elif self.isiOS:
        # <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="动卡空间" name
        #="动卡空间" label="动卡空间" enabled="true" visible="true" x="68" y="395" width="338" height
        #="26"/>
        # iOSTagStaticText = "XCUIElementTypeStaticText"
        # elementTag = element.tag
        # isStaticText = elementTag == iOSTagStaticText
        # return isStaticText
        # return self.is_element_SomeType_iOS(element, "XCUIElementTypeStaticText")

        # <XCUIElementTypeSearchField type="XCUIElementTypeSearchField" value="gh_cfcfcee0
        32cc" name="搜索" label="搜索" enabled="true" visible="true" x="41" y="27" width="319" heig
        ht="36">
            #     <XCUIElementTypeButton type="XCUIElementTypeButton" name="清除文本" label="清除
            文本" enabled="true" visible="false" x="335" y="35" width="20" height="20"/>
        # </XCUIElementTypeSearchField>
        isSearchField = self.is_element_SomeType_iOS(element, "XCUIElementTypeSearchField")

        # <XCUIElementTypeTextField type="XCUIElementTypeTextField" value="请输入手机号" lab
        el="" enabled="true" visible="true" x="107" y="188" width="265" height="24"/>
        isTextField = self.is_element_SomeType_iOS(element, "XCUIElementTypeTextField")
        #
        isSecureTextField = self.is_element_SomeType_iOS(element, "XCUIElementTypeSecureTe
        xtField")
        isEditableText = isSearchField or isTextField or isSecureTextField
        return isEditableText

def is_element_Link(self, element):
    # 元素是否是 XCUIElementTypeLink

```

```

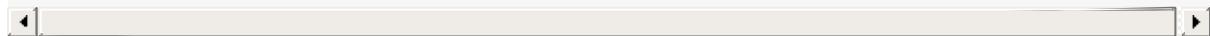
"""
    <XCUIElementTypeLink type="XCUIElementTypeLink" name="阅读原文" label="阅读原文" enabled="true" visible="false" x="16" y="1512" width="64" height="19">
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="阅读原文" name="阅读原文" label="阅读原文" enabled="true" visible="false" x="16" y="1512" width="64" height="19"/>
    </XCUIElementTypeLink>
"""

return self.is_element_SomeType_iOS(element, "XCUIElementTypeLink")

def is_element_SomeType_iOS(self, element, typeName):
    elementType = None
    if hasattr(element, "tag"):
        # lxml Element
        elementTag = element.tag
        elementType = elementTag
    elif hasattr(element, "attrs"):
        # BeautifulSoup soup node
        elementAttrDict = element.attrs
        elementType = elementAttrDict.get("type")
    isCurrentType = elementType == typeName
    return isCurrentType

def is_element_SomeType_Android(self, element, typeName):
    curClass = element.attrib.get("class")
    isTypeInClass = typeName in curClass
    isCurrentType = isTypeInClass
    return isCurrentType

```



## 点击元素（中间坐标值）

```

def clickElementCenterPosition(self, curElement):
    """Click center position of element

Args:
    curElement (Element): Beautiful soup / lxml element / wda Element
Returns:
    bool
Raises:
    """
hasClicked = False
# centerPos = None
centerX = None
centerY = None

hasBounds = hasattr(curElement, "bounds")
curBounds = None
if hasBounds:
    curBounds = curElement.bounds

```

```

if hasBounds and curBounds:
    # wda element
    if hasattr(curBounds, "center"):
        # is wda Rect
        curRect = curBounds
        rectCenter = curRect.center
        centerX = rectCenter[0]
        centerY = rectCenter[1]
    else:
        attrDict = None
        if hasattr(curElement, "attrs"):
            # Beautiful soup node
            attrDict = curElement.attrs
        elif hasattr(curElement, "attrib"):
            # lxml element
            attrDict = dict(curElement.attrib)

        if attrDict:
            logging.info("attrDict=%s", attrDict)
            hasCoordinate = ("x" in attrDict) and ("y" in attrDict) and ("width" in attrDict) and ("height" in attrDict)
            if hasCoordinate:
                x = int(attrDict["x"])
                y = int(attrDict["y"])
                width = int(attrDict["width"])
                height = int(attrDict["height"])
                centerX = x + int(width / 2)
                centerY = y + int(height / 2)

            if centerX and centerY:
                centerPos = (centerX, centerY)
                self.tap(centerPos)
                logging.info("Clicked center position: %s", centerPos)
                hasClicked = True

return hasClicked

```

调用：

```

moreInfoSoup = parentCellSoup.find(
    'XCUIElementTypeButton',
    attrs={"type": "XCUIElementTypeButton", "name": "更多信息", "enabled": "true", "visible": "true"},
)
if moreInfoSoup:
    clickedOk = self.clickElementCenterPosition(moreInfoSoup)

```

或：

```

page = self.get_page_source()
backElement, nextPage = self.findRealBackElement(page)

```

```
if backElement is not None:  
    isFoundAndClicked = self.clickElementCenterPosition(backElement)
```

或：

```
# try return to main page, by find main menu and click first main menu  
mainMenuList = self.get_elements_MainMenu(page)  
if mainMenuList:  
    firstMainMenu = mainMenuList[0]  
    clickOk = self.clickElementCenterPosition(firstMainMenu)
```

或：

```
isGetProxyTypeOk, respInfo = self.iOSLaunchSettingsAndGetProxyType()  
curProxySoup = respInfo  
curProxyAttrDict = curProxySoup.attrs  
curTypeName = curProxyAttrDict.get("value")  
  
# into config proxy page  
self.clickElementCenterPosition(curProxySoup)
```

## 电脑相关

### 获取电脑序列号

```
def getSerialNumber(self):  
    """get current computer serial number"""  
    # cmd = "wmic bios get serialnumber"  
    cmd = ""  
    if CommonUtils.osIsWindows():  
        # Windows  
        cmd = "wmic bios get serialnumber"  
    elif CommonUtils.osIsMacOS():  
        # macOS  
        cmd = "system_profiler SPHardwareDataType | awk '/Serial/ {print $4}'"  
    # TODO: add support other OS  
    # AIX: aix  
    # Linux: linux  
    # Windows/Cygwin: cygwin  
  
    serialNumber = ""  
    lines = CommonUtils.get_cmd_lines(cmd)  
    if CommonUtils.osIsWindows():  
        # Windows  
        serialNumber = lines[1]  
    elif CommonUtils.osIsMacOS():  
        # macOS  
        serialNumber = lines[0] # C02Y3N10JHC8, 'VMfvNykaZWi1'
```

```
    return serialNumber
```

调用：

```
serialNumber = self.getSerialNumber() # 'VMfvNykaZW1'
```

## 调试相关

在安卓手机测试期间，往往会遇到一些和调试相关内容，此处整理出其中相对通用部分，供参考。

### 缩放图片（到原始尺寸比例）

```
def scaleToOrginSize(self, screenshotImgPath, curScale):
    """resize to original screen size, according to session scale"""
    curScreenImg = Image.open(screenshotImgPath)
    originSize = curScreenImg.size # 750x1334
    newWidthInt = int(float(originSize[0]) / curScale)
    newHeightInt = int(float(originSize[1]) / curScale)
    scaledSize = (newWidthInt, newHeightInt) # 375x667
    scaledFile = screenshotImgPath
    CommonUtils.resizeImage(curScreenImg, newSize=scaledSize, outputImageFile=scaledFile)
    return scaledFile
```

### 获取当前屏幕截图文件

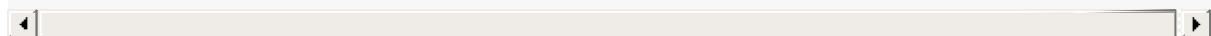
```
def getCurScreenshot(self, saveFolder=None):
    """get current screenshot image file path"""

    curDatetimeStr = CommonUtils.getCurDatetimeStr() # '20200422_144915'
    # suffix = "png"
    suffix = "jpg" # '20200422_144915.jpg'
    curFilename = "%s.%s" % (curDatetimeStr, suffix)
    if not saveFolder:
        if self.isAndroid:
            # saveFolder = self.config["CurAndroidAppScreenshotRoot"]
            # saveFolder = self.config["CurAndroidWeixinScreenshotRoot"]
            # saveFolder = self.config["debug"]["screenshot"]["Android"]["weixin"]
            saveFolder = self.config["debug"]["screenshot"]["Android"]["gameApp"] # 'debug
/Android/app/游戏app/screenshot'
        elif self.isiOS:
            # saveFolder = self.config["CuriOSWeixinScreenshotRoot"]
            # saveFolder = self.config["CuriOSAppPageSourceRoot"]
            # saveFolder = self.config["debug"]["pageSource"]["iOS"]["app"]
            # saveFolder = self.config["debug"]["screenshot"]["iOS"]["weixin"]
            # saveFolder = self.config["debug"]["screenshot"]["iOS"]["app"]
            saveFolder = self.config["debug"]["screenshot"]["iOS"]["normalApp"]
    # add current date sub folder
    curDateStr = CommonUtils.getCurDatetimeStr("%Y%m%d") # '20210107'
```

```

    saveFolder = os.path.join(saveFolder, curDateStr) # 'debug/Android/app/游戏/app/screenshots/20210107'
    CommonUtils.createFolder(saveFolder)
    fullImgFilePath = os.path.join(saveFolder, curFilename)
    beforeDriverScreenshotTime = datetime.now()
    if self.isAndroid:
        fullImgFilePath = self.driver.screenshot(fullImgFilePath) # 'debug/Android/app/游戏/app/screenshots/20210107/20210107_205117.jpg'
        # optimize size
        displayInfo = self.driver.device_info["display"] # {'width': 720, 'height': 1600}
        originSize = (displayInfo["height"], displayInfo["width"]) # (1600, 720)
        CommonUtils.resizeImage(fullImgFilePath, originSize, outputImageFile fullImgFilePath)
    elif self.isiOS:
        fullImgFilePath = self.debugIOSSaveScreenshot(saveFolder, curScale self.curSession.scale)
        afterDriverScreenshotTime = datetime.now()
        driverScreenshotTime = afterDriverScreenshotTime - beforeDriverScreenshotTime
        logging.debug("driver screenshot time: %s", driverScreenshotTime)
    return fullImgFilePath

```



## 给当前屏幕截图加标记（红框）

```

def debugDrawScreenRect(self, curRect, curImagePath None, isShow=False, isAutoSave=True, isDrawClickedPosCircle=False):
    """for debug, draw rectangle for current screenshot"""
    if not curImagePath:
        curImagePath = self.getCurScreenshot()

    curImg = CommonUtils.imageDrawRectangle(
        curImagePath,
        curRect,
        isShow isShow,
        isAutoSave isAutoSave,
        isDrawClickedPosCircle isDrawClickedPosCircle,
    )

    return curImg

```

## 给元素加边框标记

```

def debugDrawElementRect(self, elementList, curImagePath None, isShowEach False, isSaveEach=True, isDrawInSinglePic False):
    """for debug, to draw rectangle for each element in current screenshot"""
    if not curImagePath:
        curImagePath = self.getCurScreenshot()

    curImg = Image.open(curImagePath)

```

```

for eachElement in elementList:
    curBoundList = self.get_ElementBounds(eachElement)
    curWidth = curBoundList[2] - curBoundList[0]
    curHeight = curBoundList[3] - curBoundList[1]
    curRect = [curBoundList[0], curBoundList[1], curWidth, curHeight]
    curTimeStr = CommonUtils.getCurDatetimeStr("%H%M%S")
    curSaveTail = "_rect_{}_{x|y|w|h}_{format}(curTimeStr) # '_rect_155618_x|y|w|h'

    curInputImg = None
    if isDrawInSinglePic:
        curInputImg = curImg
    else:
        curInputImg = curImgPath
    curImg = CommonUtils.imageDrawRectangle(
        curInputImg,
        curRect,
        isShow isShowEach,
        isAutoSave isSaveEach,
        saveTail curSaveTail,
        isDrawClickedPosCircle False,
    )

# always save final result
curTimeStr = CommonUtils.getCurDatetimeStr("%H%M%S")
finalSaveTail = "_rect_all_{s}" % curTimeStr
imgFolderAndName, pointSuffix = os.path.splitext(curImagePath)
imgFolderAndName = imgFolderAndName + finalSaveTail
finalImgPath = imgFolderAndName + pointSuffix
curImg.save(finalImgPath)

return

```

## 保存当前截图对应的xml源码

```

def debugSaveCurPageSource(self, filePrefix="", saveFolder=None):
    """for debug, save current page source xml file"""
    savedSourceFile = None
    curDatetimeStr = CommonUtils.getCurDatetimeStr()
    sourceFormat "xml"
    # sourceFilename = "%s_source.%s" % (curDatetimeStr, sourceFormat) # '20200221_152817_
    source.xml'
    sourceFilename = "%s.%s" % (curDatetimeStr, sourceFormat)
    if filePrefix:
        sourceFilename = "%s_%s" % (filePrefix, sourceFilename)
        # 'com.netease.cloudmusic_20200221_170337.xml'

    if not saveFolder:
        # if self.isAndroid:
        #     # saveFolder = self.config["CurAndroidAppPageSourceRoot"]
        #     # saveFolder = self.config["CurAndroidWeixinPageSourceRoot"]
        #     # saveFolder = self.config["debug"]["pageSource"]["Android"]["weixin"]

```

```

#     saveFolder = self.config["debug"]["pageSource"]["Android"]["app"]
# elif self.isiOS:
#     # saveFolder = self.config["CuriOSWeixinPageSourceRoot"]
#     # saveFolder = self.config["debug"]["pageSource"]["iOS"]["weixin"]
#     saveFolder = self.config["debug"]["pageSource"]["iOS"]["app"]

# if self.isAndroid:
#     platformType = "Android"
# elif self.isiOS:
#     platformType = "iOS"
# taskType = self.taskType
# saveFolder = self.config["debug"]["pageSource"][platformType][taskType]
saveFolder = self.config["debug"]["pageSource"][self.platformType][self.taskType]

CommonUtils.createFolder(saveFolder)
sourceFilename = os.path.join(saveFolder, sourceFilename)

pageSource = self.getCurPageSource()
CommonUtils.saveTextToFile(sourceFilename, pageSource)
savedSourceFile = sourceFilename
logging.debug("saved page source: %s", savedSourceFile)
return savedSourceFile

```

## 保存当前屏幕的图片和源码

```

def debugSaveScreenAndSource(self):
    self.getCurScreenshot()
    self.debugSaveCurPageSource()

```

## 打印元素属性值

```

def debugPrintElement(self, curElement, prefix ""):
    """for debug, to print current element"""
    curElementStr = ""
    curInfoDict = {}
    keyList = []
    if self.isAndroid:
        if hasattr(curElement, "attrib"):
            curInfoDict = curElement.attrib
            keyList = ["resource-id", "class", "bounds", "text"]
        else:
            curInfoDict = curElement.info
            keyList = ["resourceName", "className", "bounds", "text"]
    elif self.isiOS:
        curInfoDict = curElement.attrib
        # keyList = ["type", "name", "label", "value", "enabled", "visible"]
        keyList = ["type", "name", "label", "value", "enabled", "visible", "x", "y", "width", "height"]

    valueList = []

```

```
for eachKey in keyList:  
    if eachKey in curInfoDict.keys():  
        eachValue = curInfoDict.get(eachKey)  
        eachValueStr = str(eachValue)  
        valueList.append(eachValueStr)  
    # else:  
    #     logging.debug("no %s key for %s", eachKey, curInfoDict)  
  
    curElementStr = " | ".join(valueList)  
    logging.info("%s element: %s", prefix, curElementStr)  
return
```

调用：

```
self.debugPrintElement(curSubElement, "is subSubLen=1")
```

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2021-07-18 09:59:23

## 实际案例

此处整理和 uiautomator2 相关的一些实际案例，供参考。

### 监听特定元素

对于如下各种常见的按钮，可以用对应代码实现自动点击：

- 确定类按钮
- 广告类弹框
- Vivo自动安装app
- 奇虎360自动登录账号

详情请见后续章节。

以及其他一些小的例子：

### 下一步

代码：

```
NextStep_Button_Xpath_List: [
    "//android.widget.TextView[@text='下一步' and contains(@resource-id, 'id/bt_dialog_right_tv')]", # 设置权限说明 启用存储权限 -> 下一步
]

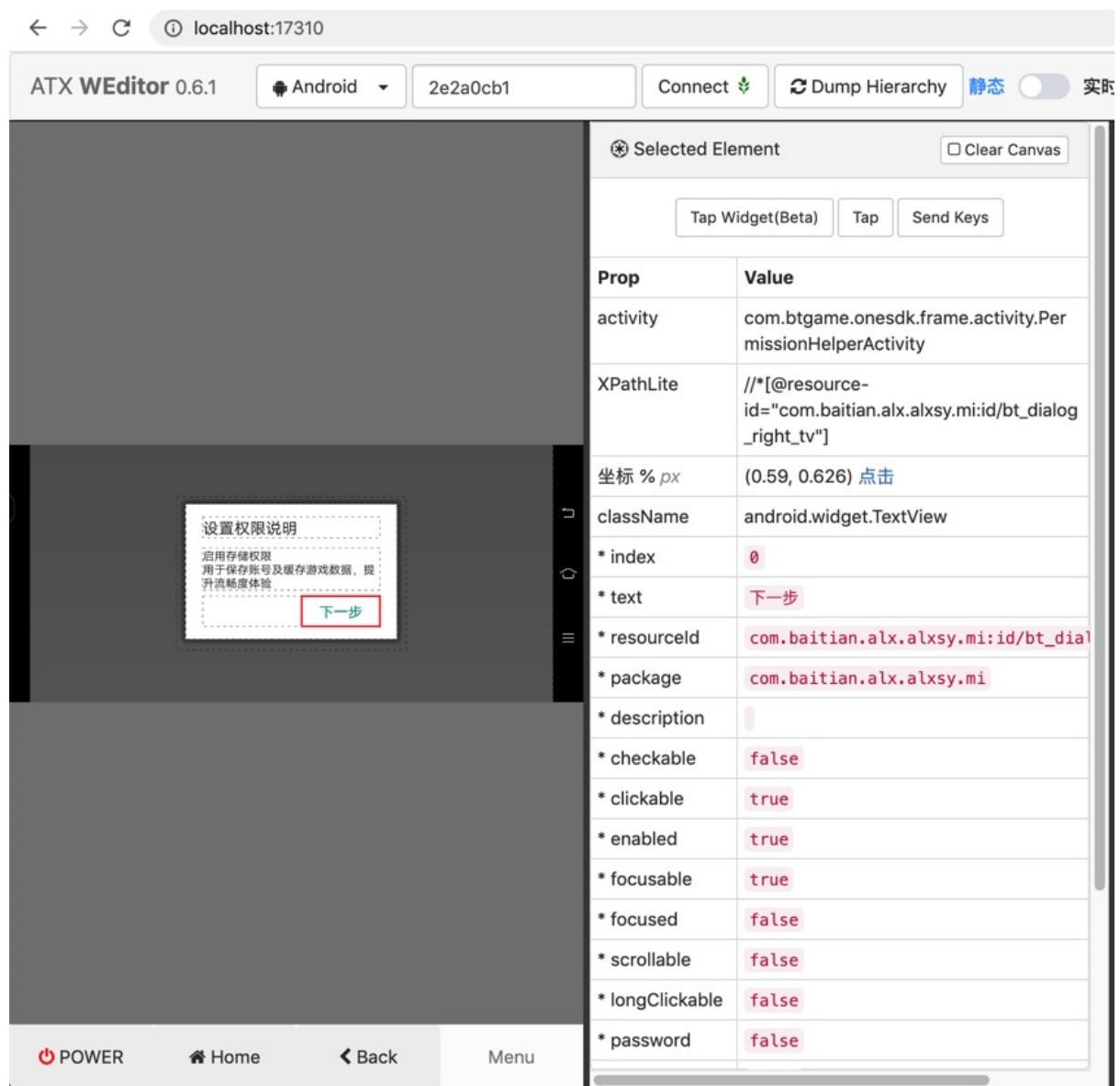
for eachXpath in NextStep_Button_Xpath_List:
    self.driver.watcher.when(eachXpath).click()
```

可以点击 下一步 类的按钮：

游戏app截图：



weditor截图：



属性：

```

Prop      Value
activity   com.btgame.onesdk.frame.activity.PermissionHelperActivity
XPathLite  //*[@resource-id="com.baitian.alx.alxsy.mi:id/bt_dialog_right_tv"]
坐标 % px  (0.59, 0.626) 点击
className  android.widget.TextView
* index    0
* text     下一步
* resourceId  com.baitian.alx.alxsy.mi:id/bt_dialog_right_tv
* package   com.baitian.alx.alxsy.mi
* description
* checkable false
* clickable true
* enabled   true
* focusable true

```

```
* focused    false
* scrollable  false
* longClickable  false
* password   false
* selected   false
# rect      {"x":815,"y":423,"width":218,"height":85}
```

详见：

【或许解决】用uiautomator2实现自动检测并点击安卓弹框：下一步

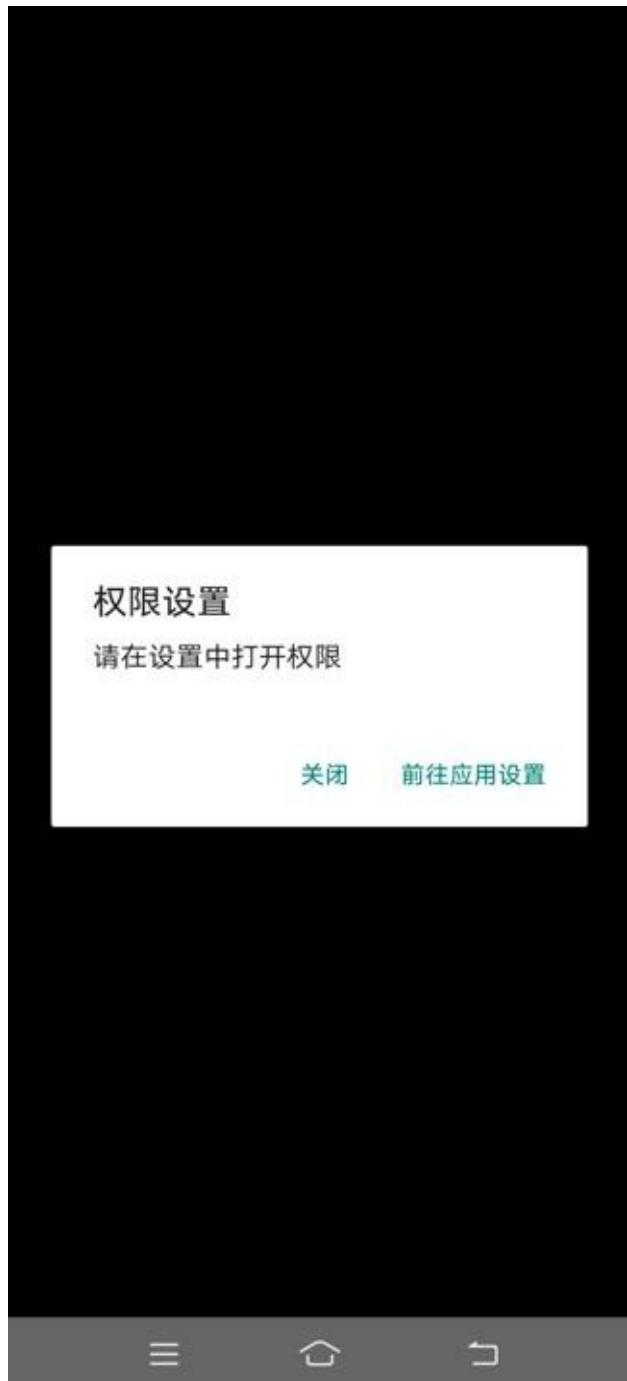
## 自动同意前往应用设置的权限

代码：

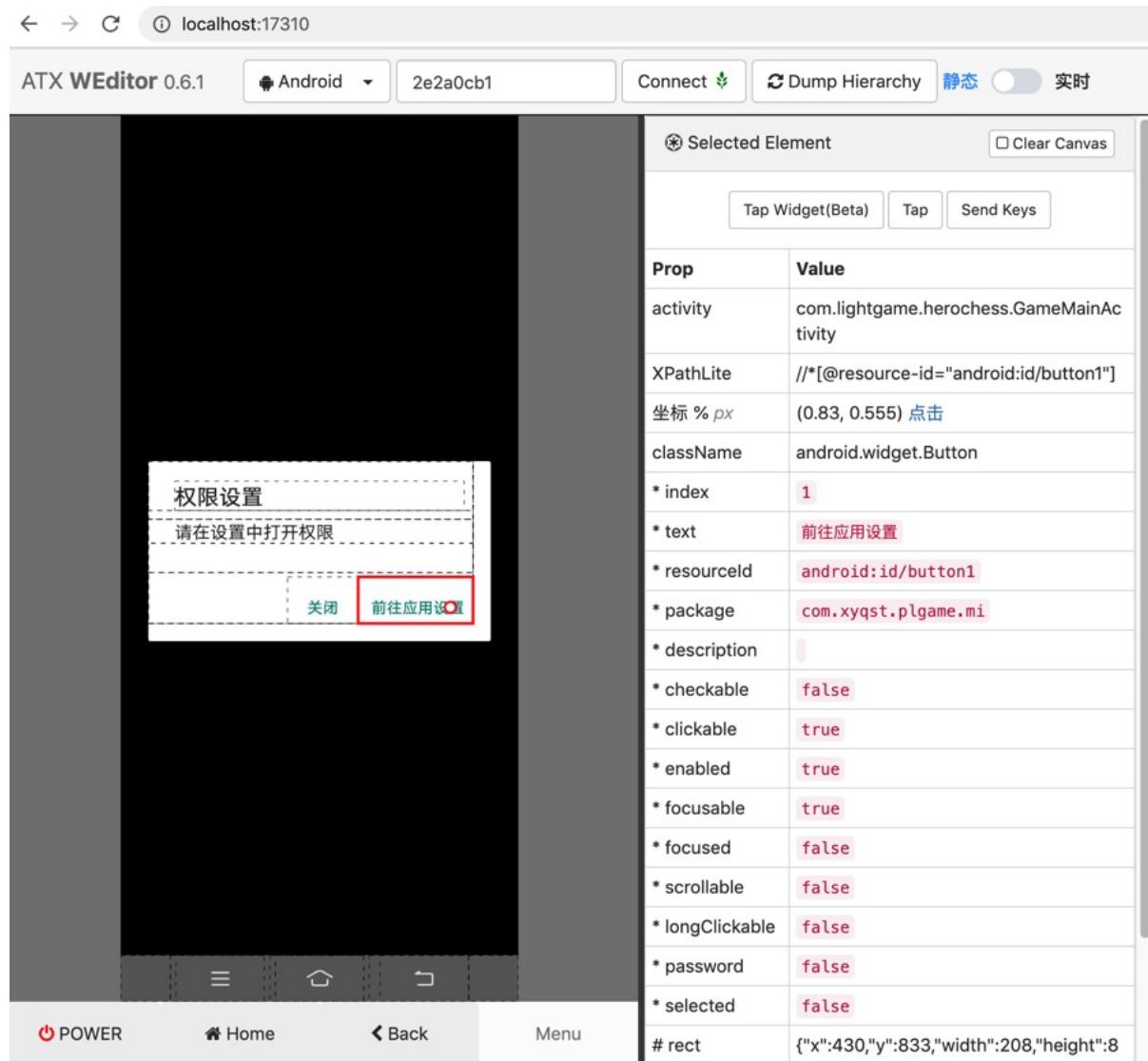
```
Permission_Settings_GoTo_App_Config_Xpath: "//android.widget.Button[@resource-id='android:id/button1' and @text='前往应用设置']"

def autoGrantPermission(self):
    """Auto grant permission for permission popup
    权限设置 弹框
    关闭 前往应用设置
    """
    gotoAppSettingsStr = "前往应用设置"
    gotoAppSettingsElement = self.driver(text gotoAppSettingsStr, className="android.widget.Button", resourceId="android:id/button1", clickable="true")
    if gotoAppSettingsElement:
        gotoAppSettingsElement.click()
        logging.info("Has clicked %s button", gotoAppSettingsStr)
        time.sleep(0.1)
    else:
        logging.warning("Not found %s button", gotoAppSettingsStr)
```

游戏app截图：



weditor截图：



属性：

```

Prop      Value
activity   com.lightgame.herochess.GameMainActivity
XPathLite  //*[@resource-id="android:id/button1"]
坐标 % px (0.83, 0.555) 点击
className  android.widget.Button
* index    1
* text     前往应用设置
* resourceId android:id/button1
* package   com.xyqst.plgame.mi
* description
* checkable false
* clickable true
* enabled   true
* focusable true
* focused   false
* scrollable false
* longClickable false
* password   false

```

```
* selected    false
# rect      {"x":430,"y":833,"width":208,"height":84}
代码 d(resourceId="android:id/button1")
```

注：未完待续

详见：

【未解决】自动化测试工具新增逻辑：权限设置弹框前往应用设置并允许

crifan.com，使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2021-07-18 09:59:23

# 百度搜索自动化

用uiautomator2去操作安卓手机，模拟实现百度搜索自动化：打开浏览器，打开百度首页，输入 crifan，点击搜索，显示搜索结果，解析搜索结果（的标题和链接地址）

## 代码

- 文件：[uiautomator2DemoBaiduSearch.py](#)
- 贴出来是

```
# Function: uiAutomator2 demo baidu search
# Author: Crifan Li
# Update: 20210417

# import time
import uiAutomator2 as u2

d = u2.connect() # connect to device
print("d.info=%s" % d.info)
# d.info={'currentPackageName': 'com.android.browser', 'displayHeight': 2201, 'displayRotation': 0, 'displaySizeDpX': 393, 'displaySizeDpY': 873, 'displayWidth': 1080, 'productName': 'atom', 'screenOn': True, 'sdkInt': 29, 'naturalOrientation': True}

# for debug: get current app info
# curApp = d.app_current()
# print("curApp=%s" % curApp)

# for debug: get running app list
# activeAppList = d.app_list_running()
# print("activeAppList=%s" % activeAppList)

#####
# Launch browser
#####
Browser_XiaomiBuiltIn = "com.android.browser"
browserPackage = Browser_XiaomiBuiltIn
# d.app_start(browserPackage)
d.app_start(browserPackage, stop=True)

# wait util browser launch complete -> appear 我的 tab
# MustShowTabName = "主页"
MustShowTabName = "我的"
# d(text=MustShowTabName).exists(timeout=10)
d(text=MustShowTabName, packageName=browserPackage).exists(timeout=10)
print("Browser homepage loaded")

#####
# Open baidu homepage
#####
```

```

SearchInputId = "com.android.browser:id/b4w"

# # open new window
# windowUiObj = d(resourceId="com.android.browser:id/dm")
# windowUiObj.click()

# # click add to new window
# addNewWindowUiObj = d(resourceId="com.android.browser:id/akr")
# addNewWindowUiObj.click()

# for debug
# curPageXml = d.dump_hierarchy(compressed=False, pretty=False)
# print("curPageXml=%s" % curPageXml)

# find input box inside address bar

# # Method 1: use driver pass in parameter
# inputUiObj = d(resourceId=SearchInputId, className="android.widget.TextView")
# # inputUiObj = d(resourceId=SearchInputId)
# print("type(inputUiObj)=%s" % type(inputUiObj)) # type(inputUiObj)=<class 'uiautomator2.session.UiObject'>
# print("inputUiObj=%s" % inputUiObj) # inputUiObj=<uiautomator2.session.UiObject object at 0x10a0bea00>
# inputUiObjectInfo = inputUiObj.info
# print("type(inputUiObjectInfo)=%s" % type(inputUiObjectInfo)) # type(inputUiObjectInfo)=<class 'dict'>
# print("inputUiObjectInfo=%s" % inputUiObjectInfo) # inputUiObjectInfo={'bounds': {'bottom': 172, 'left': 160, 'right': 797, 'top': 107}, 'childCount': 0, 'className': 'android.widget.TextView', 'contentDescription': '搜索框', 'packageName': 'com.android.browser', 'resourceName': 'com.android.browser:id/b4h', 'text': '', 'visibleBounds': {'bottom': 172, 'left': 160, 'right': 797, 'top': 107}, 'checkable': False, 'checked': False, 'clickable': True, 'enabled': True, 'focusable': False, 'focused': False, 'longClickable': False, 'scrollable': False, 'selected': False}
# isFoundInput = inputUiObj.exists # True

# # Method 2: use xpath
# inputXPathSelector = d.xpath("//android.widget.TextView[@resource-id=SearchInputId]")
# # inputXPathSelector = d.xpath("//*[@resource-id=SearchInputId]")
# print("type(inputXPathSelector)=%s" % type(inputXPathSelector)) # type(inputXPathSelector)=<class 'uiautomator2.xpath.XPathSelector'>
# inputXPathElem = inputXPathSelector.get()
# print("type(inputXPathElem)=%s" % type(inputXPathElem)) # type(inputXPathElem)=<class 'uiautomator2.xpath.XMLElement'>
# print("inputXPathElem=%s" % inputXPathElem) # inputXPathElem=<uiautomator2.xpath.XMLElement object at 0x108585d30>
# print("type(inputXPathElem.attrib)=%s" % type(inputXPathElem.attrib)) # type(inputXPathElem.attrib)=<class 'lxml.etree._Attrib'>
# print("inputXPathElem.attrib=%s" % inputXPathElem.attrib) # inputXPathElem.attrib={'index': '1', 'text': '', 'resource-id': 'com.android.browser:id/b4h', 'package': 'com.android.browser', 'content-desc': '搜索框', 'checkable': 'false', 'checked': 'false', 'clickable': 'true', 'enabled': 'true', 'focusable': 'false', 'focused': 'false', 'scrollable': 'false', 'long-clickable': 'false', 'password': 'false', 'selected': 'false', 'visible-to-user': ''

```

```

'true', 'bounds': '[160,107][797,172]'}
# isFoundInput = inputXPathSelector.exists # True

# trigger into input page

# Method 1
inputUIObj = d(resourceId SearchInputId, className "android.widget.TextView")
inputUIObj.click()
print("Clicked search box")

# # Method 2
# inputXPathSelector = d.xpath("//android.widget.TextView[@resource-id=%s]" % SearchInputI
d)
# inputXPathSelector.click()

# input baidu homr url
BaiduHomeUrl = "https://www.baidu.com/"
AddressInputId = "com.android.browser:id/bqi"
searchUIObj = d(resourceId AddressInputId, className="android.widget.EditText")
searchUIObj.set_text(BaiduHomeUrl)
print("Inputed baidu homepage url: %s" % BaiduHomeUrl)

# trigger jump to baidu home
EnterKey = "enter"
d.press(EnterKey)
print("Emulated press key %s" % EnterKey)

# wait util baidu home loaded
# d(text="百度一下", resourceId="com.android.browser:id/bq3").exists(timeout=10)
d(text="百度一下,你就知道", className="android.view.View").exists(timeout 10)
print("Baidu home loaded")

#####
# Input text
#####
searchStr = "crifan"

baiduSearchKeywordUIObj = d(resourceId "index-kw", className "android.widget.EditText")
baiduSearchKeywordUIObj.set_text(searchStr)
print("Inputed baidu search text %s" % searchStr)

#####
# Trigger baidu search
#####

# # Method 1: press key
# TriggerSearchKey = "enter" # work
# # TriggerSearchKey = "search" # not work
# # TriggerSearchKey = "go" # not work
# # TriggerSearchKey = "done" # not work
# d.press(TriggerSearchKey)
# print("Emulated press key %s" % TriggerSearchKey)

```

```

# Method 2: find 百度一下 button then click
baiduSearchButtonUiObj = d(resourceId "index-bn", className "android.widget.Button")
baiduSearchButtonUiObj.click()
print("Clicked baidu search button")

#####
# Extract search result content
#####

# Special: for fixbug of get page xml is not latest, so using following code to refresh to
# get latest page source xml
d.service("uiautomator").stop()
d.service("uiautomator").start()
# time.sleep(1)

# for debug
# get page source xml
# curPageXml = d.dump_hierarchy(compressed=False, pretty=False)
# print("curPageXml=%s" % curPageXml)
# with open("baidu_search_%s_result_pageSource_reloaded.xml" % searchStr, "w") as fp:
#     fp.write(curPageXml)

d(resourceId "results").exists(timeout 10)

# Note: following syntax can NOT find elements
# resultsSelector = d.xpath("//*[@resource-id='results']")
# titleButtonSelectorList = resultsSelector.xpath("//android.widget.Button[@clickable='true']").all()
# titleButtonSelectorList = resultsSelector.xpath("./android.widget.Button[@clickable='true']").all()

# Xpath chain search can find elements
titleButtonElementList = d.xpath("//*[@resource-id='results']/android.widget.Button[@clickable='true']").all()
titleButtonNum = len(titleButtonElementList)
print("Found %s search result title" % titleButtonNum)

# descriptionElementList = d.xpath("//*[@resource-id='results']/android.view.View[1]/android.view.View[1]/android.view.View[2]/android.view.View[1]/android.view.View[1]/android.view.View[1]/android.view.View[1]/android.view.View[1]/android.view.View[1]").all()
descriptionElementList = d.xpath("//*[@resource-id='results']/android.view.View/android.view.View[1]/android.view.View[2]/android.view.View[1]/android.view.View[1]/android.view.View[1]/android.view.View[1]/android.view.View[1]/android.view.View[1]").all()
descriptionNum = len(descriptionElementList)
print("Found %s description" % descriptionNum)

# # sourceWebsiteElementList = d.xpath('//*[@resource-id="results"]/android.view.View/android.view.View[1]/android.view.View[2]/android.view.View[1]').all()
# sourceWebsiteElementList = d.xpath('//*[@resource-id="results"]/android.view.View/android.view.View[1]/android.view.View[2]/android.view.View[2]').all()
# sourceWebsiteNum = len(sourceWebsiteElementList)
# print("Found %s source website" % sourceWebsiteNum)

```

```
for curIdx, eachTitleButtonElement in enumerate(titleButtonElementList):
    curNum = curIdx + 1
    print("%s [%d/%d] %s" % ("-"*20, curNum, titleButtonNum, "-"*20))
    # eachTitleButtonElemAttrib = eachTitleButtonElement.attrib
    # print("title attrib: %s" % eachTitleButtonElemAttrib)
    # curTitle = eachTitleButtonElemAttrib["text"]
    curTitle = eachTitleButtonElement.text
    print("title=%s" % curTitle)

    curDescriptionElem = descriptionElementList[curIdx]
    curDescription = curDescriptionElem.text
    print("description=%s" % curDescription)

    # curSourceWebsiteElem = sourceWebsiteElementList[curIdx]
    # curSourceWebsite = curSourceWebsiteElem.text
    # print("curSourceWebsite=%s" % curSourceWebsite)

print("Demo baidu search complete")
```

实现了：

- 打开安卓中浏览器，进入首页
- 输入百度首页地址，进入百度首页
- 给输入框中输入crifan，并触发搜索
- 等待页面加载完毕
  - (由于手机屏幕有限，当前页面只显示了3个搜索结果)
- 获取并解析每一条的标题title和description描述
- 并打印输出

视频：



截图：

The screenshot shows the PyCharm IDE interface with the following details:

- Project Structure:** The project is named "BaiduSearchAutomation".
- Code Editor:** The file "uiautomator2DemoBaiduSearch.py" is open, containing Python code for automating a Baidu search. The code uses the `UiAutomator2` library to interact with the Android browser.
- Terminal:** The Python Debug Console shows the execution of the script. It prints the current package name as "com.android.browser" and the device information as "Device: Redmi, orga4pmzee4ts47". It also shows the process ID (pid) and left coordinate (left) for various UI elements like the search bar and results.
- Output:** The terminal output includes logs from the `UiAutomator2` library, such as "Found 3 search result title" and "Found 3 description". It also shows the search results for "crifan - Bing 词典 - 必应" and "crifan - 词典 - CSDN博客 - crifan".
- Watch:** A watch point is set on the variable "parent" of an XPATH object.
- Breakpoints:** Breakpoints are set at line 175 and line 176 of the script.
- Run Configuration:** The configuration is set to "Python: 当前文件" (Current File).

输出：

```
d.info {'currentPackageName': 'com.android.browser', 'displayHeight': 2201, 'displayRotation': 0, 'displaySizeDpX': 393, 'displaySizeDpY': 873, 'displayWidth': 1080, 'productName': 'Atom', 'screenOn': True, 'sdkInt': 29, 'naturalOrientation': True}
Browser homepage loaded
Clicked search box
Inputed baidu homepage url: https://www.baidu.com/
Emulated press key enter
Baidu home loaded
Inputed baidu search text crifan
Clicked baidu search button
[D 210409 23:03:10 __init__:764] restart-uiautomator since "UiAutomation not connected"
[D 210409 23:03:11 __init__:817] Device: Redmi, orga4pmzee4ts47t
[D 210409 23:03:11 __init__:823] stop app: com.github.uiautomator
[D 210409 23:03:11 __init__:827] kill process(ps): uiautomator
[D 210409 23:03:11 __init__:840] uiautomator-v2 is starting ... left: 40.0s
[D 210409 23:03:12 __init__:840] uiautomator-v2 is starting ... left: 39.0s
[D 210409 23:03:13 __init__:840] uiautomator-v2 is starting ... left: 38.0s
[D 210409 23:03:14 __init__:840] uiautomator-v2 is starting ... left: 36.9s
[D 210409 23:03:15 __init__:840] uiautomator-v2 is starting ... left: 35.9s
[D 210409 23:03:16 __init__:840] uiautomator-v2 is starting ... left: 34.9s
[D 210409 23:03:17 __init__:840] uiautomator-v2 is starting ... left: 33.8s
[I 210409 23:03:17 __init__:784] uiautomator back to normal
Found 3 search result title
Found 3 description
----- [1/3] -----
title 在路上on the way - 走别人没走过的路,让别人有路可走
description crifan 4周前 (03-09) 177浏览 0评论 google 收录查询谷歌收录查询,如何让Google收录网站
- 云...
----- [2/3] -----
```

```

title crifan简介 - crifan的专栏 - CSDN博客 - crifan
description 2020年2月15日crifan简介 各种计算机语言简介和总结 v1.0 08-03 本文主要介绍了各种计算机
语言的分类和关系,以及记录了笔者对其中部分语言,如Word VBA,Excel...
----- [3/3] -----
title crifan - Bing 词典 - 必应
description crifan网络目录;下午;我个人 网络释义 1. 目录 如何实现Linux下的U盘驱动 v0.4 - . . .
Specification 规范 crifan 目录 Mass Storage ...

```

## 调试

期间用weditor调试页面，找元素属性。

比如：

找浏览器中 百度一下 的标题：



拷贝出元素属性值：

| Prop     | Value            |
|----------|------------------|
| activity | .BrowserActivity |

```
package com.android.browser
className android.widget.EditText
description
text 百度一下
resourceId com.android.browser:id/bq3
clickable true
label
x, y 160, 83
width 659
height 112
enabled true
visible
XPath //*[@resource-id="com.android.browser:id/bq3"]
position (0.275, 0.064) 点击
代码 d(resourceId="com.android.browser:id/bq3")
```

其中（常可用于定位元素的）核心参数是：

- resourceId : com.android.browser:id/bq3
- text : 百度一下
- className : android.widget.EditText

即可写出，比如，判断该元素是否出现的代码了：

```
d(text "百度一下", resourceId "com.android.browser:id/bq3").exists(timeout 10)
```

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2021-07-18 09:59:23

# 微信相关

## 查找微信公众号中文全名

代码：

```
# def findWeixinPublicAccountZhcnSoup(self, soup, curAccountId):
def findWeixinPublicAccountZhcnFullName(self, soup, curAccountId):
    """Find weixin public account element's zh-CN full name

    Args:
        soup (soup): soup of current page xml
    Returns:
        public account zh-CN full name
    Raises:
    """

# accountZhcnTextSoup = None
accountZhcnFullName = ""
parentNodeLocator = None

"""

    搜索结果中文名节点是Text
    <XCUIElementTypeOther type="XCUIElementTypeOther" name="搜一搜" label="搜一搜" enabled="true" visible="true" x="0" y="70" width="414" height="666">
        <XCUIElementTypeOther type="XCUIElementTypeOther" value="2" name="公众号" label="公众号" enabled="true" visible="true" x="16" y="83" width="398" height="23">
            <XCUIElementTypeOther type="XCUIElementTypeOther" value="2" enabled="true" visible="true" x="16" y="83" width="52" height="23">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="2" name="公众号" label="公众号" enabled="true" visible="true" x="16" y="83" width="52" height="22"/>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="16" y="129" width="60" height="60"/>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="88" y="126" width="310" height="23">
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="动卡空间" name="动卡空间" label="动卡空间" enabled="true" visible="true" x="88" y="126" width="70" height="22"/>
        </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="88" y="151" width="310" height="21">
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="分享用卡活动, 传播正能量" name="分享用卡活动, 传播正能量" label="分享用卡活动, 传播正能量" enabled="true" visible="true" x="88" y="152" width="184" height="19"/>
        </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="88" y="178" width="266" height="20">
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="微信" name="微信" label="微信" enabled="true" visible="true" x="88" y="179" width="266" height="20"/>
        </XCUIElementTypeOther>
    </XCUIElementTypeOther>
</XCUIElementTypeOther>
```

```

号: " name="微信号: " label="微信号: " enabled="true" visible="true" x="88" y="178" width="58" height="19"/>
    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="gh_cfcfcee032cc" name="gh_cfcfcee032cc" label="gh_cfcfcee032cc" enabled="true" visible="true" x="145" y="178" width="112" height="19"/>
</XCUIElementTypeOther>

```

搜索结果中文名节点是Other，其下是多个Text节点：

```

<XCUIElementTypeOther type="XCUIElementTypeOther" name="搜一搜" label="搜一搜" enabled="true" visible="true" x="0" y="70" width="375" height="600">
    <XCUIElementTypeOther type="XCUIElementTypeOther" value="2" name="公众号" label="公众号" enabled="true" visible="true" x="16" y="83" width="359" height="23">
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="2" name="公众号" label="公众号" enabled="true" visible="true" x="16" y="83" width="52" height="23"/>
    </XCUIElementTypeOther>
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="16" y="129" width="60" height="60">
        <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="343" y="128" width="16" height="16"/>
        <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="324" y="128" width="16" height="16"/>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="88" y="126" width="271" height="23">
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="牛尔" name="牛尔" label="牛尔" enabled="true" visible="true" x="88" y="126" width="35" height="22"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="Tmall" name="Tmall" label="Tmall" enabled="true" visible="true" x="122" y="126" width="41" height="22"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="旗舰店" name="旗舰店" label="旗舰店" enabled="true" visible="true" x="162" y="126" width="53" height="22"/>
        </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="88" y="151" width="271" height="21">
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="牛尔亲研天猫官方旗舰店" name="牛尔亲研天猫官方旗舰店" label="牛尔亲研天猫官方旗舰店" enabled="true" visible="true" x="88" y="152" width="169" height="19"/>
        </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="88" y="178" width="271" height="19">
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="微信号: " name="微信号: " label="微信号: " enabled="true" visible="true" x="88" y="178" width="58" height="19"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="niuer-tmall" name="niuer-tmall" label="niuer-tmall" enabled="true" visible="true" x="145" y="178" width="70" height="18"/>
        </XCUIElementTypeOther>
    
```

公众号中文名全部是绿色的：

```

<XCUIElementTypeOther type="XCUIElementTypeOther" name="搜一搜" label="搜一搜" enabled="true" visible="true" x="0" y="70" width="414" height="666">

```

```

        <XCUIElementTypeOther type="XCUIElementTypeOther" value="2" name="公众号" label="公众号" enabled="true" visible="true" x="16" y="83" width="398" height="23">
            <XCUIElementTypeOther type="XCUIElementTypeOther" value="2" enabled="true" visible="true" x="16" y="83" width="52" height="23">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="2" name="公众号" label="公众号" enabled="true" visible="true" x="16" y="83" width="52" height="22"/>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="16" y="129" width="60" height="60"/>
            <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="382" y="128" width="16" height="16"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="88" y="126" width="310" height="23">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="limi_里美" name="limi_里美" label="limi_里美" enabled="true" visible="true" x="88" y="126" width="61" height="22"/>
            </XCUIElementTypeOther>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="88" y="151" width="310" height="21">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="真实好成分，呵护少女肌。" name="真实好成分，呵护少女肌。" label="真实好成分，呵护少女肌。" enabled="true" visible="true" x="88" y="152" width="184" height="19"/>
            </XCUIElementTypeOther>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="88" y="178" width="310" height="19">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="微信号：" name="微信号：" label="微信号：" enabled="true" visible="true" x="88" y="178" width="58" height="18"/>
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="limi_official" name="limi_official" label="limi_official" enabled="true" visible="true" x="145" y="178" width="72" height="18"/>
            </XCUIElementTypeOther>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="0" width="0" height="0"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="237" width="414" height="25">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="没有更多的搜索结果" name="没有更多的搜索结果" label="没有更多的搜索结果" enabled="true" visible="true" x="133" y="239" width="148" height="20"/>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    """
    foundAccountId = soup.find(
        'XCUIElementTypeStaticText',
        attrs {"value": curAccountId, "name": curAccountId, "type": "XCUIElementTypeStaticText"}),
    )
    logging.debug("foundAccountId=%s", foundAccountId)
    # foundAccountId=<XCUIElementTypeStaticText enabled="true" height="18" label="gh_cfcfce032cc" name="gh_cfcfcfcee032cc" type="XCUIElementTypeStaticText" value="gh_cfcfcfcee032cc" visible="true" width="112" x="145" y="178"/>

```

```

if foundAccountId:
    idParent = foundAccountId.parent
    logging.debug("idParent=%s", idParent)
    if idParent:
        # method 1: two prev.prev
        # # idParentPrev = idParent.previous_sibling
        # idParentPrev = idParent.previous_sibling.previous_sibling
        # accountDescNode = idParentPrev
        # logging.info("accountDescNode=%s", accountDescNode) # '\n'
        # if accountDescNode:
        #     # accountZhcnNode = accountDescNode.previous_sibling
        #     accountZhcnNode = accountDescNode.previous_sibling.previous_sibling
        #     logging.info("accountZhcnNode=%s", accountZhcnNode)

        # # method 2: siblings[-2] of XCUIElementTypeOther
        # idParentPrevSiblingList = idParent.previous_siblings

        # accountDescNode = None
        # accountZhcnNode = None

        # TypeOther = "XCUIElementTypeOther"
        # typeOtherNodeCurIdx = 0
        # AccountDescNodeIdx = 1
        # AccountZhcnNodeIdx = 2

        # for eachPrevSiblingNode in idParentPrevSiblingList:
        #     curNodeName = eachPrevSiblingNode.name
        #     isTypeOtherNode = curNodeName == TypeOther
        #     if isTypeOtherNode:
        #         typeOtherNodeCurIdx += 1

        #         if AccountDescNodeIdx == typeOtherNodeCurIdx:
        #             accountDescNode = eachPrevSiblingNode
        #         elif AccountZhcnNodeIdx == typeOtherNodeCurIdx:
        #             accountZhcnNode = eachPrevSiblingNode

        #     hasFoundAll = accountDescNode and accountZhcnNode
        #     if hasFoundAll:
        #         break

        # logging.info("accountDescNode=%s", accountDescNode)
        # logging.info("accountZhcnNode=%s", accountZhcnNode)

        # if accountZhcnNode:
        #     accountZhcnTextSoup = accountZhcnNode.find(
        #         'XCUIElementTypeStaticText',
        #         attrs={ "type": "XCUIElementTypeStaticText" },
        #     )

        # method 3: parent.parent is 搜一搜, direct child 2nd XCUIElementTypeOther of e
        nabled="true" visible="true"
        idParentParent = idParent.parent
        if idParentParent:

```

```

        otherSoupList = idParentParent.find_all(
            "XCUIElementTypeOther",
            attrs {"type": "XCUIElementTypeOther", "enabled":"true", "visible":"true"}),
        recursive=False,
    )
    if otherSoupList and (len(otherSoupList) >= 2):
        firstOtherSoup = otherSoupList[0]
        if firstOtherSoup.attrs["name"] == "公众号":
            secondOtherSoup = otherSoupList[1]
            zhcnNameSoupList = secondOtherSoup.find_all(
                "XCUIElementTypeStaticText",
                attrs {"type": "XCUIElementTypeStaticText", "enabled":"true",
"visible":"true"},)
            if zhcnNameSoupList:
                for eachTextSoup in zhcnNameSoupList:
                    curPartName = eachTextSoup.attrs.get("value")
                    accountZhcnFullName += curPartName
        if accountZhcnFullName:
            secondOtherAttrDict = secondOtherSoup.attrs
            parentX = secondOtherAttrDict["x"]
            parentY = secondOtherAttrDict["y"]
            parentWidth = secondOtherAttrDict["width"]
            parentHeight = secondOtherAttrDict["height"]
            parentNodeLocator = {
                "type": "XCUIElementTypeOther",
                "enabled": "true",
                "visible": "true",
                "x": parentX,
                "y": parentY,
                "width": parentWidth,
                "height": parentHeight,
            }
    }

# return accountZhcnTextSoup
# return accountZhcnFullName
return accountZhcnFullName, parentNodeLocator

```



支持多种情况：

- 普通的：动卡空间

- • 中英文混合：牛尔Tmall旗舰店

◦

- 中英文混合带绿色的： limi里美

◦



## 确定类的按钮

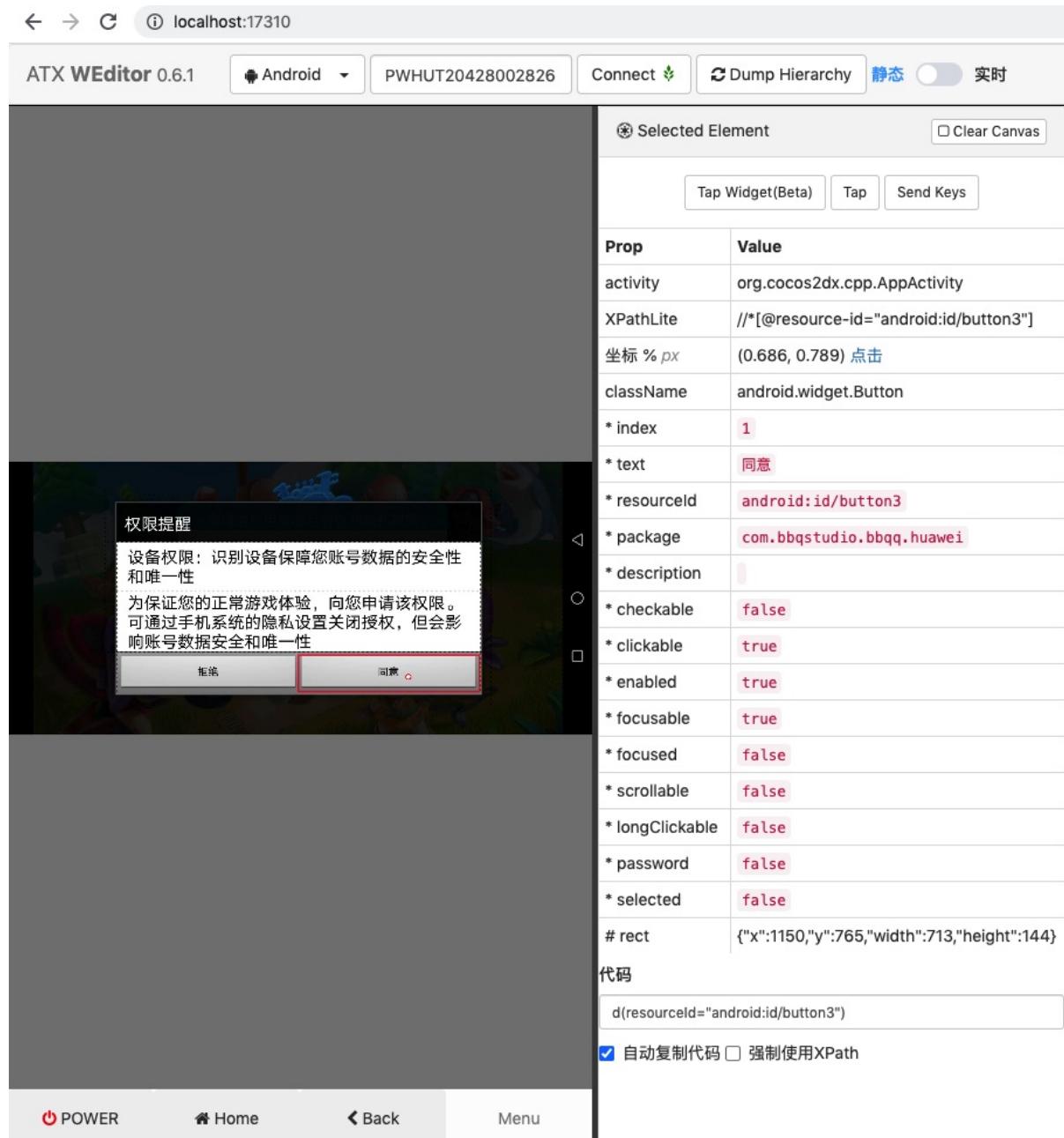
代码：

```
# 注：此处 确定 按钮，加了这么多 属性判断，目的是为了防止误触发其他情况，比如vivo的账号登录弹框中的确定
Confirm_Button_Xpath_List: [
    "//android.widget.Button[@text='确定' and @resource-id='android:id/button2' and @index='0' and @enabled='true']", # 权限申请 确定
    "//android.widget.Button[@text='确认' and @resource-id='android:id/button1' and @index='0' and @enabled='true']", # 提醒：权限授权确认 为了更好的游戏体验。。。 -> 确定
    "//android.widget.Button[@text='确定' and @resource-id='android:id/button1' and @index='0' and @enabled='true']", # 应用权限运行需要被授予读取外部存储。。。 -> 确定
    "//android.widget.Button[@text='确定' and @resource-id='android:id/button1' and @index='1' and @enabled='true' and @clickable='true']", # 客户端更新。。。 -> 确定
    "//android.widget.Button[@text='确定' and @resource-id='android:id/button1' and @index='2' and @enabled='true']", # 亲爱的玩家。。。 申请一些权限。。。 -> 确定
    "//android.widget.Button[@text='确定' and contains(@resource-id, 'id/continue_button') and @index='1' and @enabled='true' and @clickable='true']", # 请选择要向...授予的权限-> 确定
    com.android.permissioncontroller:id/continue_button
    "//android.widget.Button[@text='确定' and @resource-id='android:id/button3' and @index='0' and @enabled='true' and @clickable='true']", # 重要提醒。。。 点击确定接收我们的条款与隐私保护政策 确定 服务条款
    "//android.widget.Button[@text='同意' and @resource-id='android:id/button1' and @index='2' and @enabled='true' and @clickable='true']", # 为保证您正常的游戏体验。。。 -> 确定
    "//android.widget.Button[contains(@text, '知道了') and @resource-id='android:id/button1' and @index='0' and @clickable='true']", # 提示 游戏需要以下权限才能访问。。。 -> 知道了
    "//android.widget.Button[@text='同意' and contains(@resource-id, 'id/gasdk_base_privacy_agree') and @index='1' and @enabled='true' and @clickable='true']", # 隐私保护提示。。。 请点击同意 -> 确定
    "//android.widget.Button[@text='同意' and @resource-id='android:id/button3' and @index='1' and @enabled='true' and @clickable='true']", # 权限提醒 设备权限。。。 拒绝 同意
    "//android.widget.LinearLayout[contains(@resource-id, 'id/uni_p_accept_ll') and @index='1' and @enabled='true']", # 网易 用户协议和隐私政策 拒绝 同意
    "# //android.widget.Button[@text='同意并继续' and @index='2' and @clickable='true']", # 同意并继续 //*[@resource-id="com.zsfz.zwnzjsw3.vivo:string/destroy_ad_bn"]
    "//android.widget.Button[@text='同意并继续' and @clickable='true']", # 同意并继续 //*[@resource-id="com.zsfz.zwnzjsw3.vivo:string/destroy_ad_bn"]
    "//android.widget.Button[@text='同意继续' and @clickable='true']", # com.sy4399.zmxyol.vivo/造梦西游OL-新职业 稍后再说 同意继续 d(text="同意继续")
    "//android.widget.TextView[@text='同意并继续' and @clickable='true']", # 某游戏 弹框 放弃并退出 同意并继续
]
for eachXpath in Confirm_Button_Xpath_List:
    self.driver.watcher.when(eachXpath).click()
```

自动识别和点击：弹框中各种常见的确定类的按钮

## 权限提醒

weditor截图：



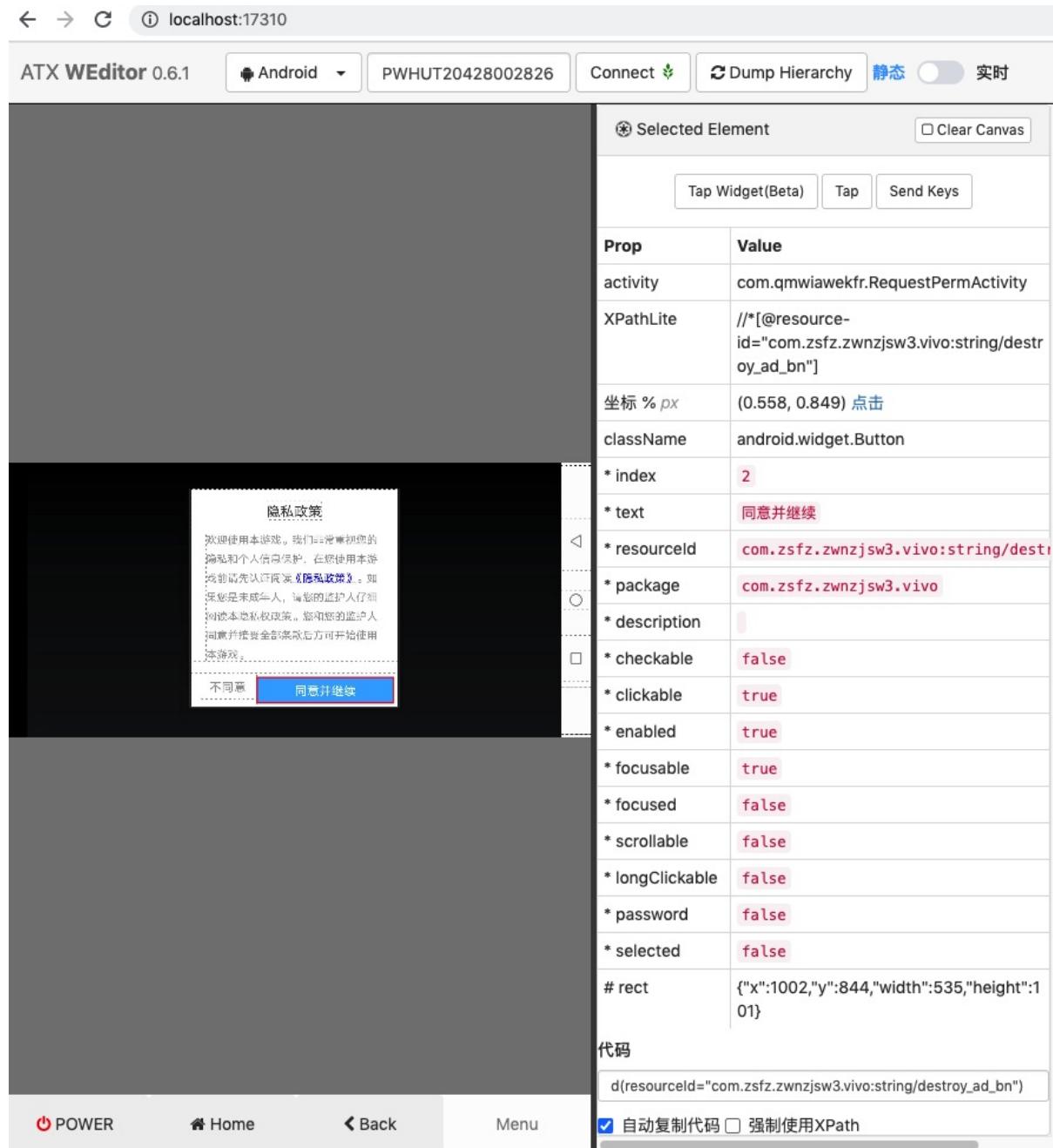
属性：

| Prop         | Value                                  |
|--------------|--|
| activity     | org.cocos2dx.cpp.AppActivity           |
| XPathLite    | //*[@resource-id="android:id/button3"] |
| 坐标 % px      | (0.686, 0.789) 点击                      |
| className    | android.widget.Button                  |
| * index      | 1                                      |
| * text       | 同意                                     |
| * resourceId | android:id/button3                     |
| * package    | com.bbqstudio.bbqq.huawei              |

```
* description
* checkable    false
* clickable    true
* enabled      true
* focusable    true
* focused      false
* scrollable   false
* longClickable false
* password     false
* selected     false
# rect {"x":1150,"y":765,"width":713,"height":144}
代码 d(resourceId="android:id/button3")
```

## 隐私政策 同意并继续

weditor截图：



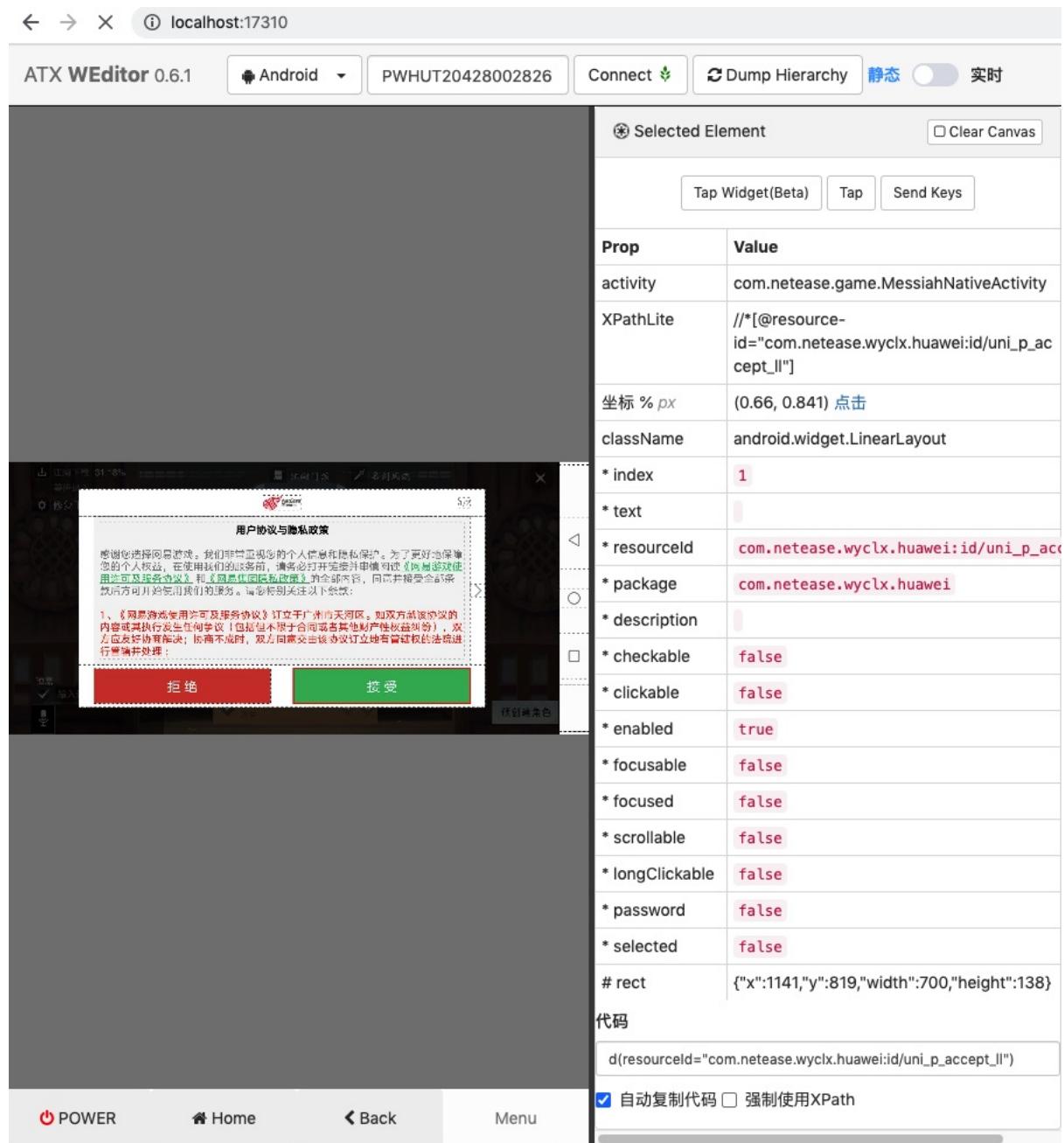
属性:

| Prop          | Value   |
|---------------|---|
| activity      | com.qmwiawekfr.RequestPermActivity                              |
| XPathLite     | //*[@resource-id="com.zsfz.zwnzjsw3.vivo:string/destroy_ad_bn"] |
| 坐标 % px       | (0.558, 0.849) 点击   |
| className     | android.widget.Button   |
| * index       | 2   |
| * text        | 同意并继续   |
| * resourceId  | com.zsfz.zwnzjsw3.vivo:string/destri                            |
| * package     | com.zsfz.zwnzjsw3.vivo  |
| * description |   |
| * checkable   | false   |
| * clickable   | true  |

```
* enabled      true
* focusable    true
* focused     false
* scrollable   false
* longClickable false
* password    false
* selected    false
# rect      {"x":1002,"y":844,"width":535,"height":101}
代码 d(resourceId="com.zsfz.zwnzjsw3.vivo:string/destroy_ad_bn")
```

## 网易的 用户协议与隐私政策 接收

游戏 com.netease.wyclx.huawei/一梦江湖 的 weditor 截图：



属性：

```
Prop      Value
activity   com.netease.game.MessiahNativeActivity
XPathLite  //*[@resource-id="com.netease.wyclx.huawei:id/uni_p_accept_11"]
坐标 % px (0.66, 0.841) 点击
className  android.widget.LinearLayout
* index    1
* text
* resourceId com.netease.wyclx.huawei:id/uni_p_accept_11
* package   com.netease.wyclx.huawei
* description
* checkable  false
* clickable  false
* enabled    true
* focusable  false
* focused    false
* scrollable false
* longClickable false
* password   false
* selected   false
# rect     {"x":1141,"y":819,"width":700,"height":138}
代码 d(resourceId="com.netease.wyclx.huawei:id/uni_p_accept_11")
```

## 个人信息保护引导 同意并继续

app截图：

Selected Element

| Prop            | Value  |
|-----------------|--|
| activity        | .activity.SplashActivity                                 |
| XPathLite       | //*[@resource-id="mobi.htjsq.accelerator:id/comfirm_tv"] |
| 坐标 % px         | (0.64, 0.78) 点击  |
| className       | android.widget.TextView                                  |
| * index         | 2  |
| * text          | 同意并继续  |
| * resourceId    | mobi.htjsq.accelerator:id/comfirm_tv                     |
| * package       | mobi.htjsq.accelerator                                   |
| * description   |  |
| * checkable     | false  |
| * clickable     | true   |
| * enabled       | true   |
| * focusable     | true   |
| * focused       | false  |
| * scrollable    | false  |
| * longClickable | false  |
| * password      | false  |
| * selected      | false  |
| # rect          | {"x":542,"y":1738,"width":430,"height":141}              |

代码

```
d(resourceId="mobi.htjsq.accelerator:id/comfirm_tv")
```

自动复制代码  强制使用XPath

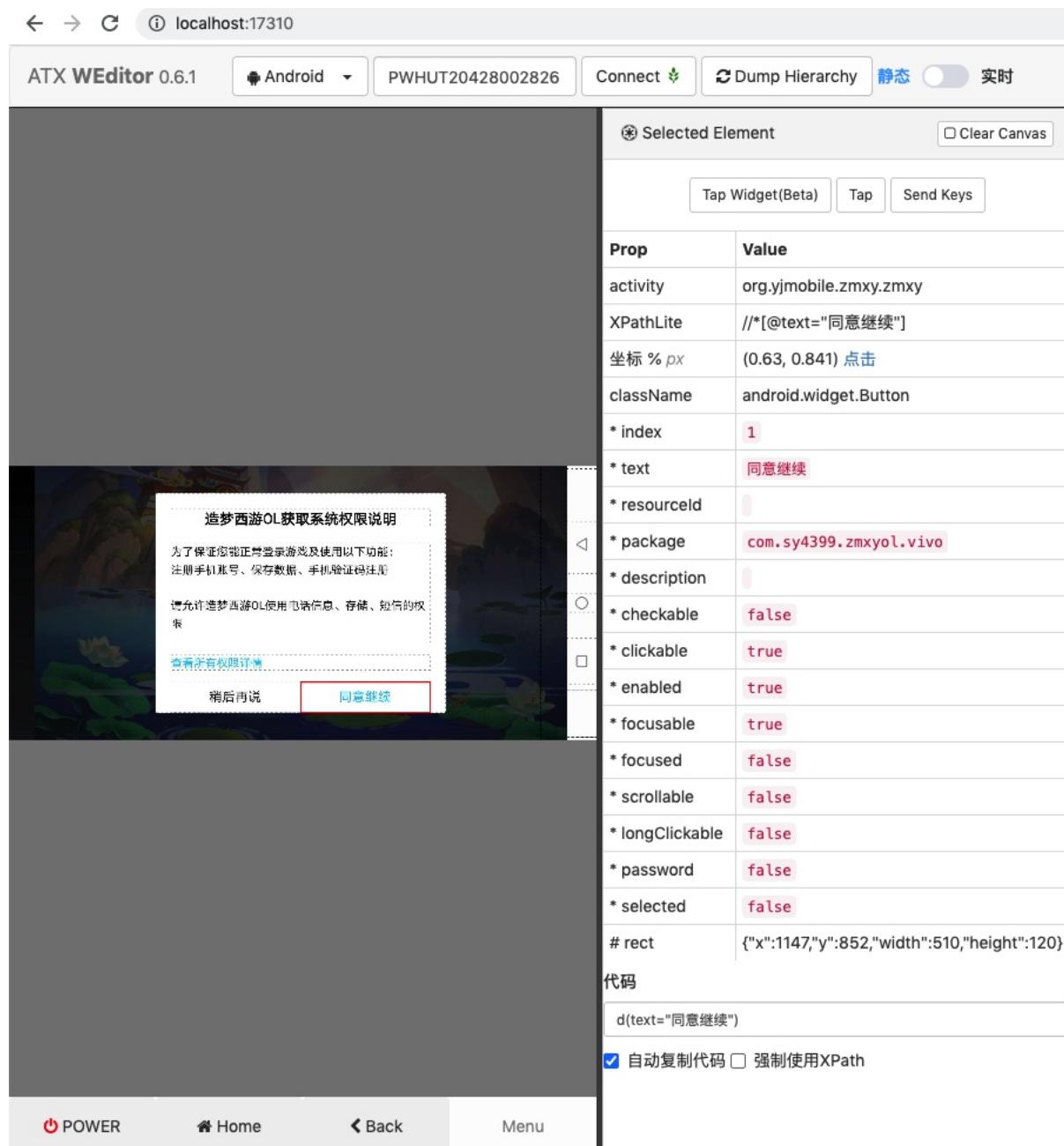
属性：

| Prop          | Value  |
|---------------|--|
| activity      | .activity.SplashActivity                                 |
| XPathLite     | //*[@resource-id="mobi.htjsq.accelerator:id/comfirm_tv"] |
| 坐标 % px       | (0.64, 0.78) 点击  |
| className     | android.widget.TextView                                  |
| * index       | 2  |
| * text        | 同意并继续  |
| * resourceId  | mobi.htjsq.accelerator:id/comfirm_tv                     |
| * package     | mobi.htjsq.accelerator                                   |
| * description |  |
| * checkable   | false  |
| * clickable   | true   |
| * enabled     | true   |

```
* focusable    true
* focused     false
* scrollable   false
* longClickable false
* password    false
* selected    false
# rect      {"x":542,"y":1738,"width":430,"height":141}
代码 d(resourceId="mobi.htjsq.accelerator:id/comfirm_tv")
```

## 获取系统权限说明 同意继续

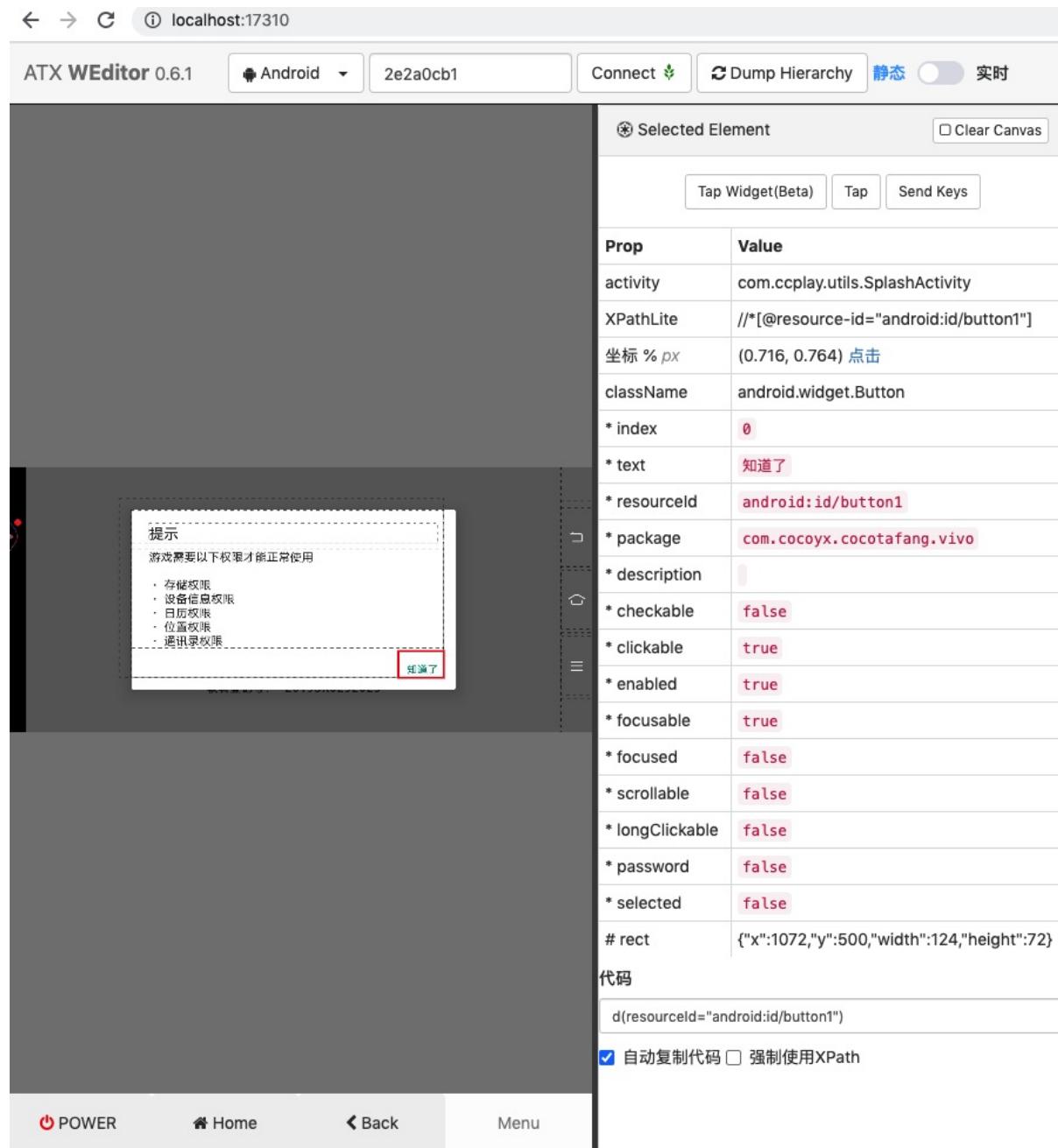
游戏 com.sy4399.zmxyol.vivo/造梦西游OL-新职业 的weditor截图：



## 提示 需要权限 知道了

游戏 com.cocoyx.cocotafang.vivo/战争模拟器

weditor截图：



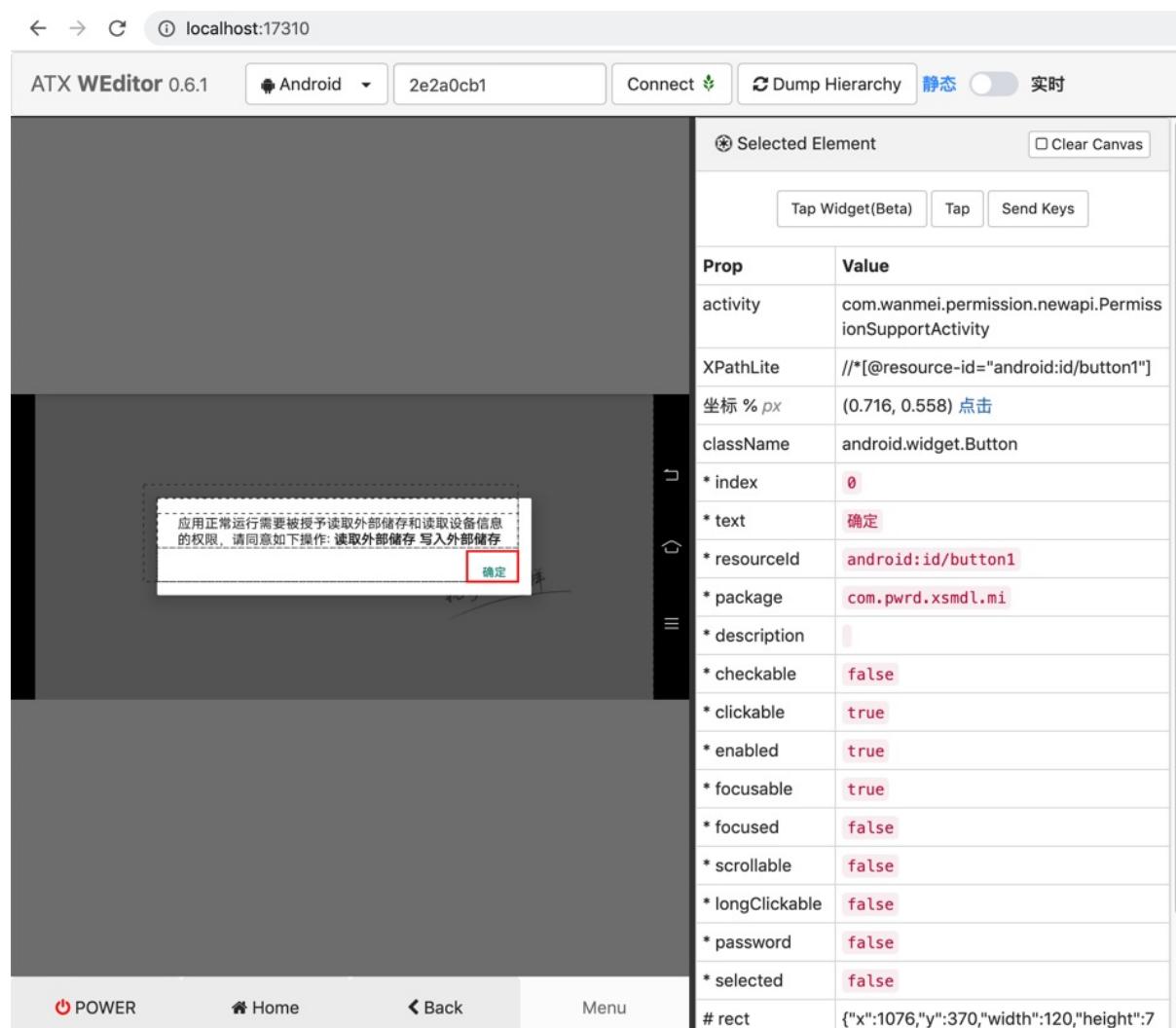
属性：

| Prop      | Value                                  |
|-----------|--|
| activity  | com.ccplay.utils.SplashActivity        |
| XPathLite | //*[@resource-id="android:id/button1"] |
| 坐标 % px   | (0.716, 0.764) 点击                      |
| className | android.widget.Button                  |
| * index   | 0                                      |
| * text    | 知道了                                    |

```
* resourceId android:id/button1
* package com.cocoxyx.cocotafang.vivo
* description
* checkable false
* clickable true
* enabled true
* focusable true
* focused false
* scrollable false
* longClickable false
* password false
* selected false
# rect {"x":1072,"y":500,"width":124,"height":72}
代码 d(resourceId="android:id/button1")
```

## 应用正常运行需要被授予 权限 确定

weditor截图：

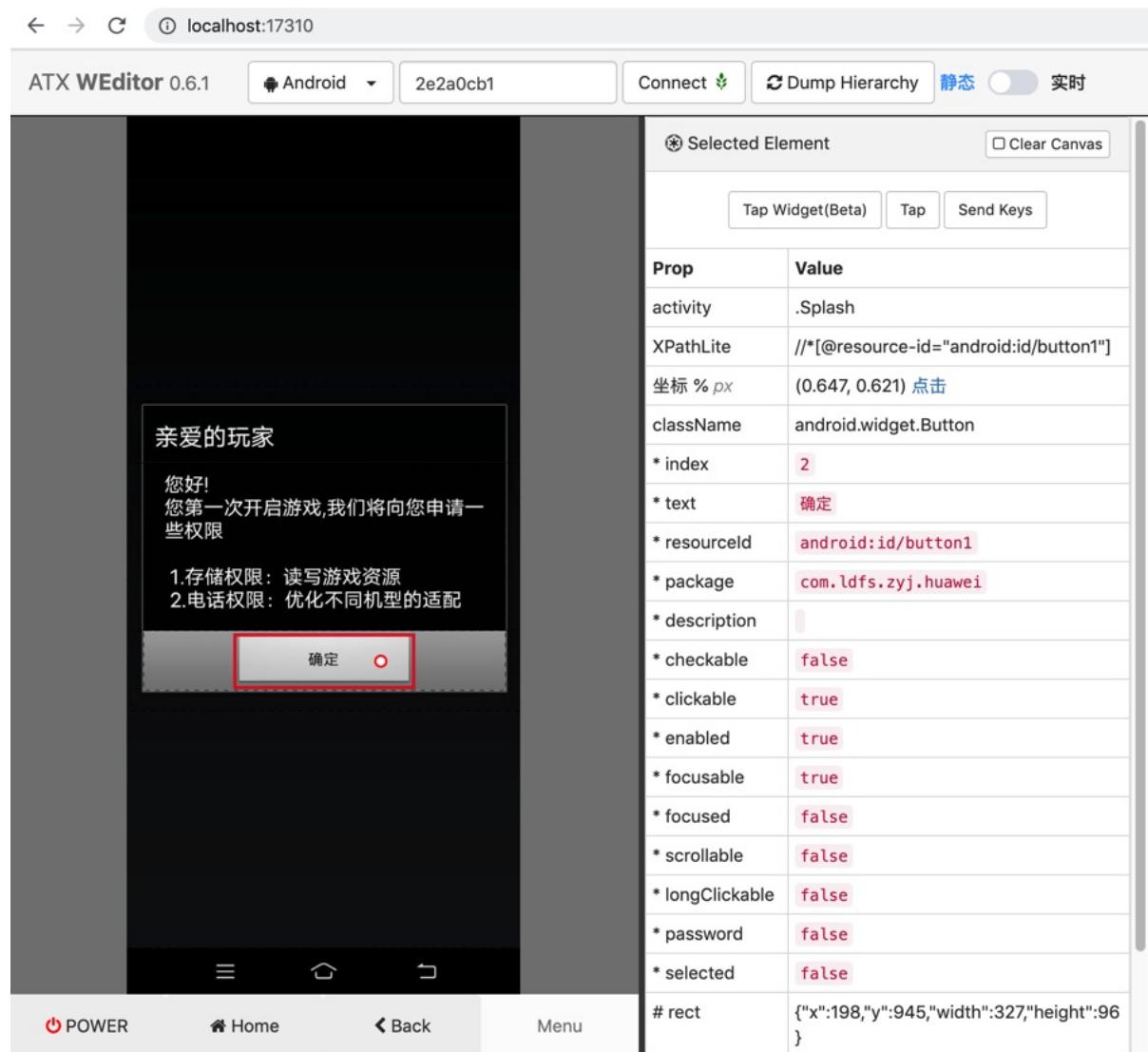


属性：

```
Prop      Value
activity   com.wanmei.permission.newapi.PermissionSupportActivity
XPathLite  //*[@resource-id="android:id/button1"]
坐标 % px  (0.716, 0.558) 点击
className  android.widget.Button
* index    0
* text     确定
* resourceId  android:id/button1
* package    com.pwrd.xsmdl.mi
* description
* checkable  false
* clickable  true
* enabled    true
* focusable  true
* focused    false
* scrollable false
* longClickable  false
* password   false
* selected   false
# rect     {"x":1076,"y":370,"width":120,"height":72}
代码
d(resourceId "android:id/button1")
```

## 亲爱的玩家 申请一些权限 确定

weditor截图：



属性：

```

Prop      Value
activity   .Splash
XPathLite  //*[@resource-id="android:id/button1"]
坐标 % px (0.647, 0.621) 点击
className  android.widget.Button
* index    2
* text     确定
* resourceId  android:id/button1
* package   com.ldfs.zyj.huawei
* description
* checkable false
* clickable true
* enabled   true
* focusable true
* focused   false
* scrollable false
* longClickable false
* password  false

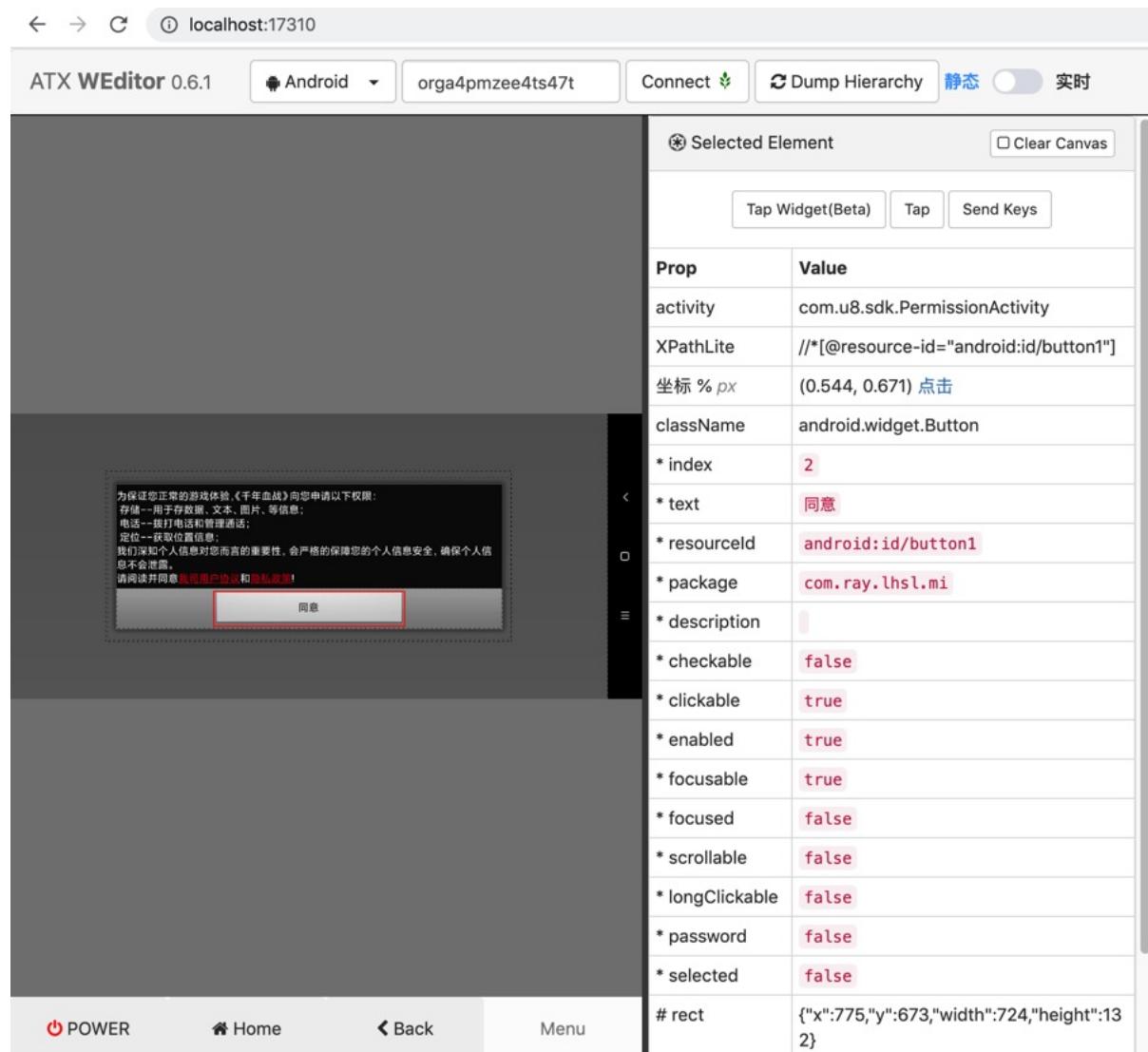
```

```
* selected false
# rect {"x":198,"y":945,"width":327,"height":96}
代码 d(resourceId="android:id/button1")
```

## 为保证您正常的游戏体验 申请权限 同意

游戏：千年血战

weditor截图：



属性：

```
Prop Value
activity com.u8.sdk.PermissionActivity
XPathLite //*[@resource-id="android:id/button1"]
坐标 % px (0.544, 0.671) 点击
className android.widget.Button
* index 2
* text 同意
```

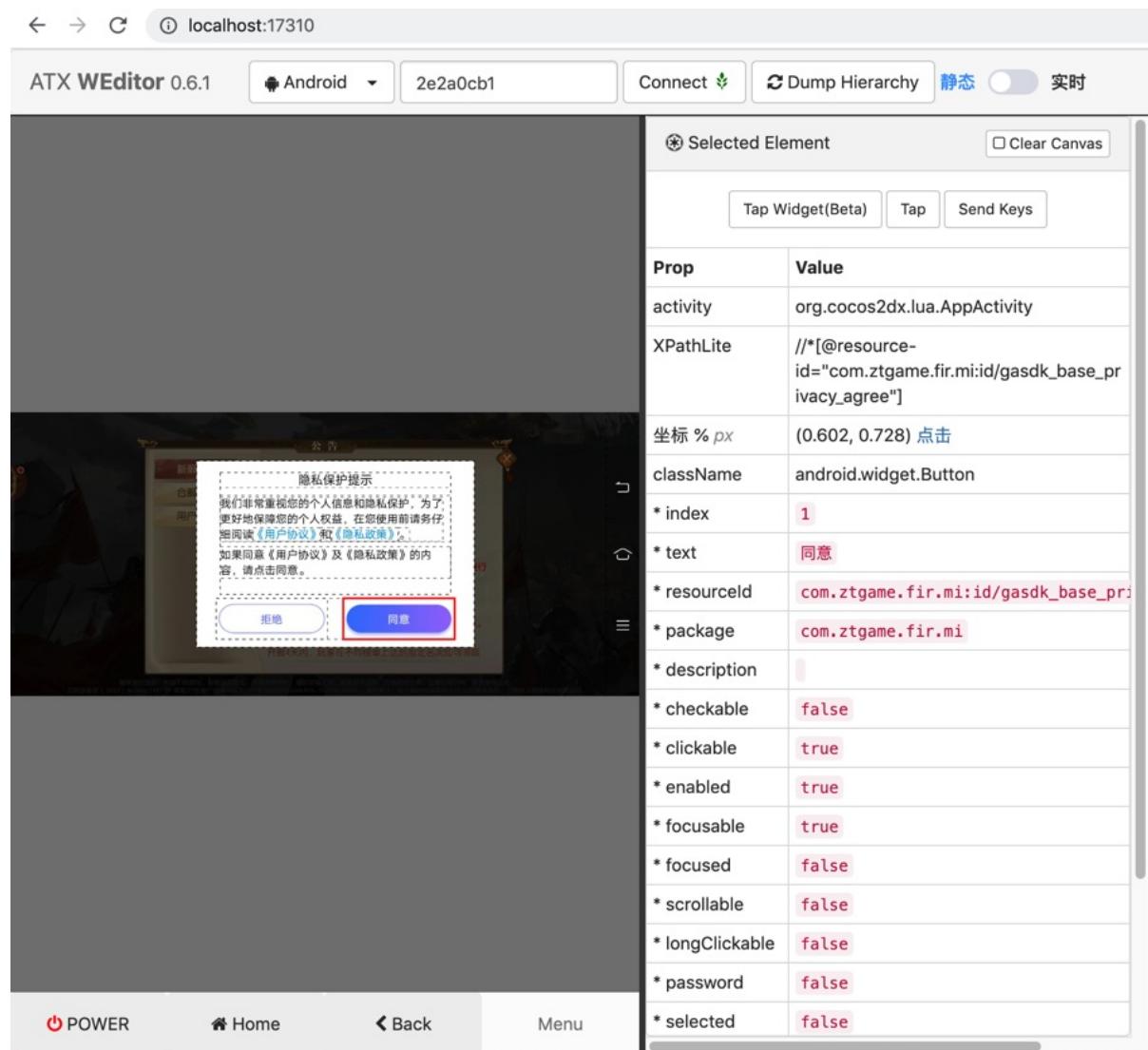
```

* resourceId    android:id/button1
* package      com.ray.lhs1.mi
* description
* checkable    false
* clickable    true
* enabled      true
* focusable    true
* focused      false
* scrollable   false
* longClickable false
* password     false
* selected     false
# rect      {"x":775,"y":673,"width":724,"height":132}
代码 d(resourceId="android:id/button1")

```

## 隐私保护提示 同意

weditor截图：

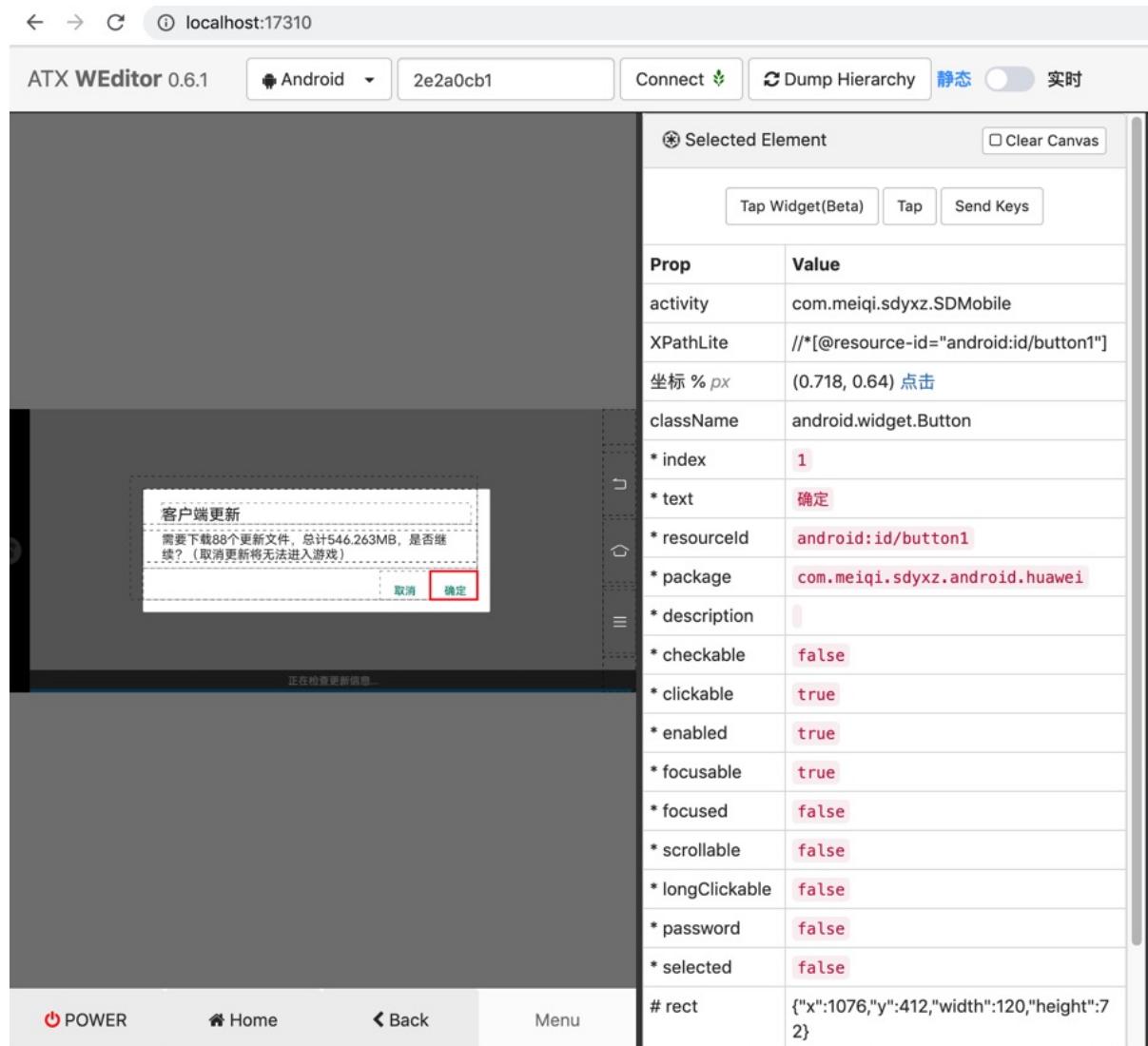


属性：

```
Prop      Value
activity   org.cocos2dx.lua.AppActivity
XPathLite  //*[@resource-id="com.ztgame.fir.mi:id/gasdk_base_privacy_agree"]
坐标 % px (0.602, 0.728) 点击
className  android.widget.Button
* index    1
* text     同意
* resourceId com.ztgame.fir.mi:id/gasdk_base_privacy_agree
* package   com.ztgame.fir.mi
* description
* checkable false
* clickable true
* enabled   true
* focusable true
* focused   false
* scrollable false
* longClickable false
* password   false
* selected   false
# rect    {"x":848,"y":472,"width":283,"height":104}
代码 d(resourceId="com.ztgame.fir.mi:id/gasdk_base_privacy_agree")
```

## 客户端更新 确定

weditor截图：



属性：

```

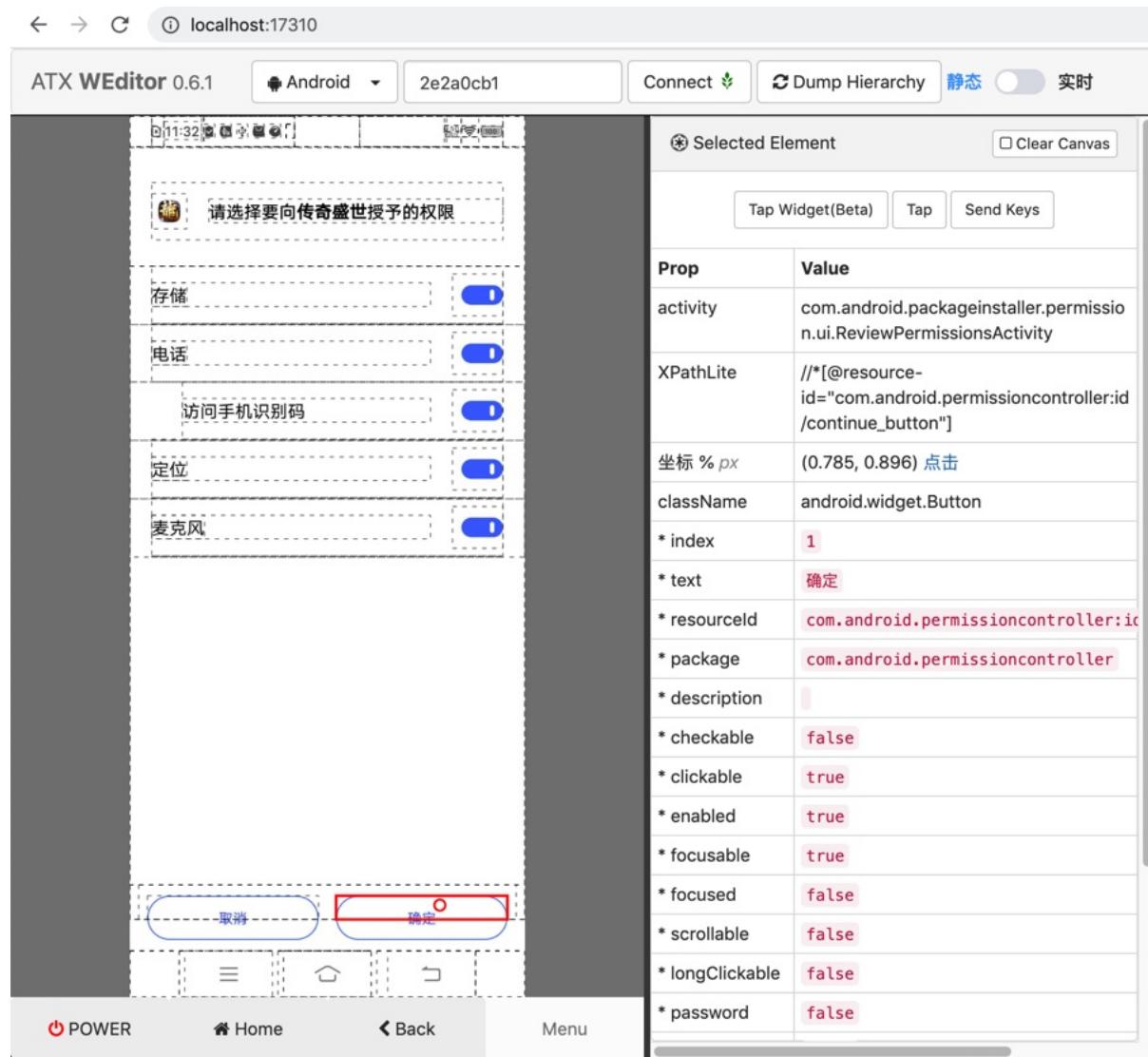
Prop      Value
activity   com.meiqi.sdyxz.SDMobile
XPathLite  //*[@resource-id="android:id/button1"]
坐标 % px  (0.718, 0.64) 点击
className  android.widget.Button
* index    1
* text     确定
* resourceId  android:id/button1
* package   com.meiqi.sdyxz.android.huawei
* description
* checkable false
* clickable true
* enabled   true
* focusable true
* focused   false
* scrollable false
* longClickable false
* password  false
* selected  false

```

```
# rect {"x":1076,"y":412,"width":120,"height":72}
代码 d(resourceId="android:id/button1")
```

## 请选择要向授予权限 确定

weditor截图：



属性：

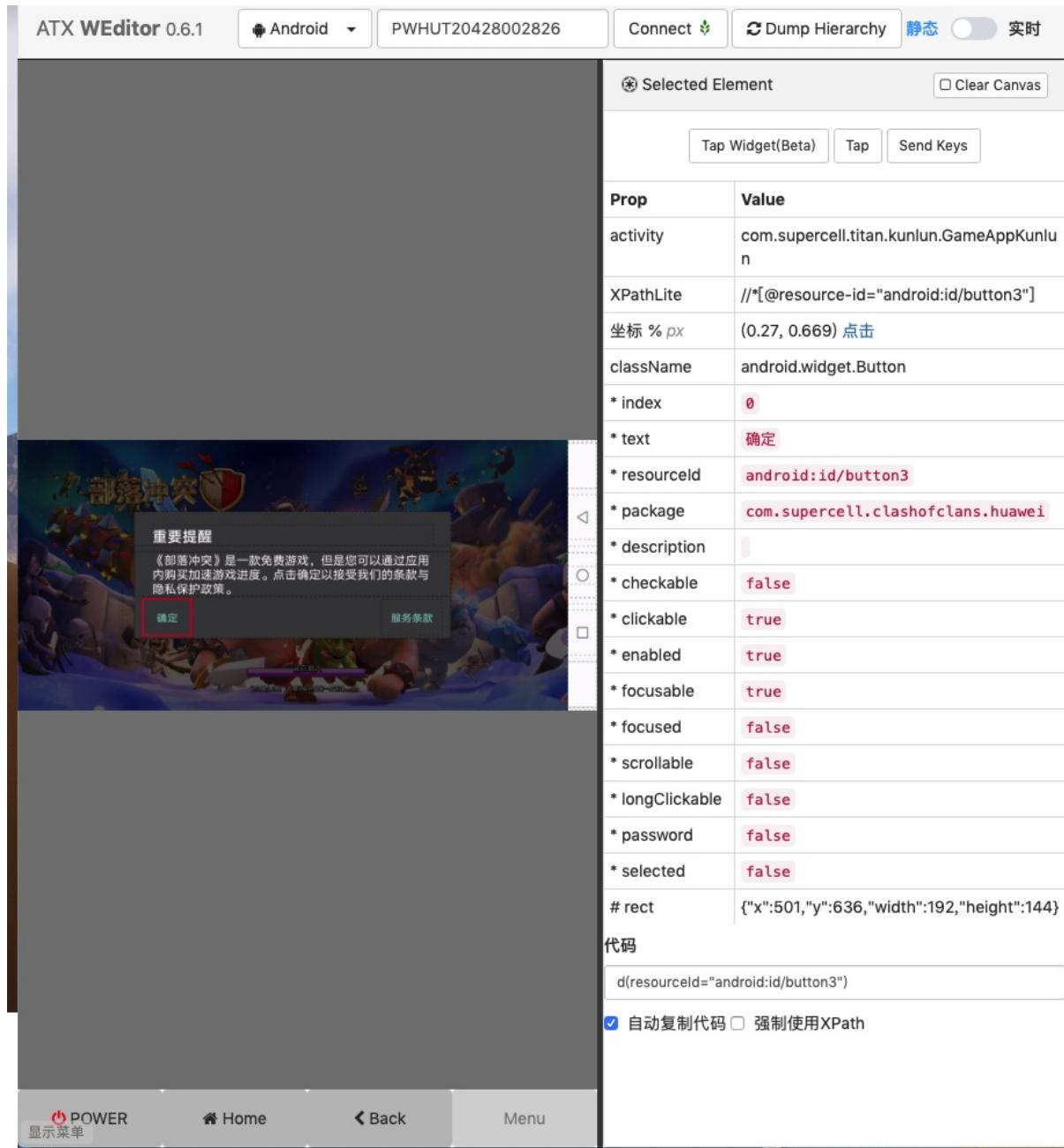
```
Prop      Value
activity   com.android.packageinstaller.permission.ui.ReviewPermissionsActivity
XPathLite  //*[@resource-id="com.android.permissioncontroller:id/continue_button"]
坐标 % px  (0.785, 0.896) 点击
className  android.widget.Button
* index    1
* text     确定
* resourceId  com.android.permissioncontroller:id/continue_button
* package   com.android.permissioncontroller
* description
```

```
* checkable    false
* clickable   true
* enabled     true
* focusable   true
* focused    false
* scrollable  false
* longClickable false
* password   false
* selected   false
# rect  {"x":376,"y":1416,"width":312,"height":43}
代码 d(resourceId="com.android.permissioncontroller:id/continue_button")
```

## 重要提醒 接受条款 确定

游戏：部落冲突

weditor截图：



属性:

| Prop          | Value                                    |
|---------------|--|
| activity      | com.supercell.titan.kunlun.GameAppKunlun |
| XPathLite     | //*[@resource-id="android:id/button3"]   |
| 坐标 % px       | (0.27, 0.669) 点击                         |
| className     | android.widget.Button                    |
| * index       | 0  |
| * text        | 确定                                       |
| * resourceId  | android:id/button3                       |
| * package     | com.supercell.clashofclans.huawei        |
| * description |  |
| * checkable   | false                                    |
| * clickable   | true                                     |
| * enabled     | true                                     |

```
* focusable    true
* focused     false
* scrollable   false
* longClickable false
* password    false
* selected    false
# rect      {"x":501,"y":636,"width":192,"height":144}
```

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2021-07-18 09:59:23

## (各大应用市场授权后的) 广告类弹框

代码:

```

PopupWindow_CloseButton_Xpath_List: [
    "//android.widget.RelativeLayout[contains(@resource-id, 'id/notice_close_view')]", # 华
为应用市场登录后 广告 弹框 关闭按钮
    "//android.widget.ImageView[contains(@resource-id, 'id/h6') and @clickable='true']", # 
vivo应用市场登录后 广告 弹框 关闭按钮 com.vivo.sdkplugin:id/h6
    "//android.widget.ImageView[contains(@resource-id, 'id/h8') and @clickable='true']", # 
vivo应用市场登录后 广告 弹框 关闭按钮 com.vivo.sdkplugin:id/h8
    "//android.widget.ImageView[contains(@resource-id, 'id/vivo_acts_mutitxt_dialog_close'
) and @clickable='true']", # vivo应用市场登录后 广告 弹框 关闭按钮 com.vivo.sdkplugin:id/vivo
_acts_mutitxt_dialog_close
    "//android.widget.ImageView[contains(@resource-id, 'id/vivo_acts_loginpic_dialog_close
') and @clickable='true']", # vivo应用市场登录后 广告 弹框 关闭按钮 com.vivo.sdkplugin:id/vivo
_acts_loginpic_dialog_close
    "//android.widget.ImageView[contains(@resource-id, 'id/vivo_login_acts_close_button')
and @clickable='true']", # vivo应用市场登录后 广告 弹框 关闭按钮 com.vivo.sdkplugin:id/vivo_l
ogin_acts_close_button
    "//android.widget.ImageView[contains(@resource-id, 'id/d6') and @clickable='true']", # 
vivo应用市场登录后 广告 弹框 关闭按钮 com.vivo.sdkplugin:id/d6
    "# //android.widget.Image[@text='7cWwAAAABJRU5ErkJggg==' and @index='0' and @clickable
='true']", # 小米市场登录后 广告 弹框 关闭按钮 d(text="7cWwAAAABJRU5ErkJggg==")
    # 小米市场登录后 广告 弹框 关闭按钮
    # 1. d(text="7cWwAAAABJRU5ErkJggg==")
    # 2. d(text="2d7mODgAAAAAAAAAAAAACV+wGa61esTL2CSwAAAABJRU5ErkJggg==")
    "# //android.widget.Image[contains(@text, 'ggg==') and @clickable='true']",
    # 防止误判 实名认证期间输入身份证时的清楚关闭按钮
    # XPathLite    //*[@text="Iuk6V5zR2fE3Srww7HUG1CXdpxkywloAa4frPB9WqgzyMteU5acplr5nrJ4F
oCqV3IQITokJqobnpSAsAoXgb4YtAy9yIXD2z1gX9dfKaFm44EagAAAABJRU5ErkJggg=="]
    # 改为:
    "//android.widget.ImageView[contains(@text, 'wAAAABJRU5ErkJggg==') and @clickable='true']",
    "//android.view.View[@text='知道了' and @index='0' and @clickable='true']", # 新人福利专
享 5元 知道了
]

for eachXpath in PopupWindow_CloseButton_Xpath_List:
    self.driver.watcher.when(eachXpath).click()

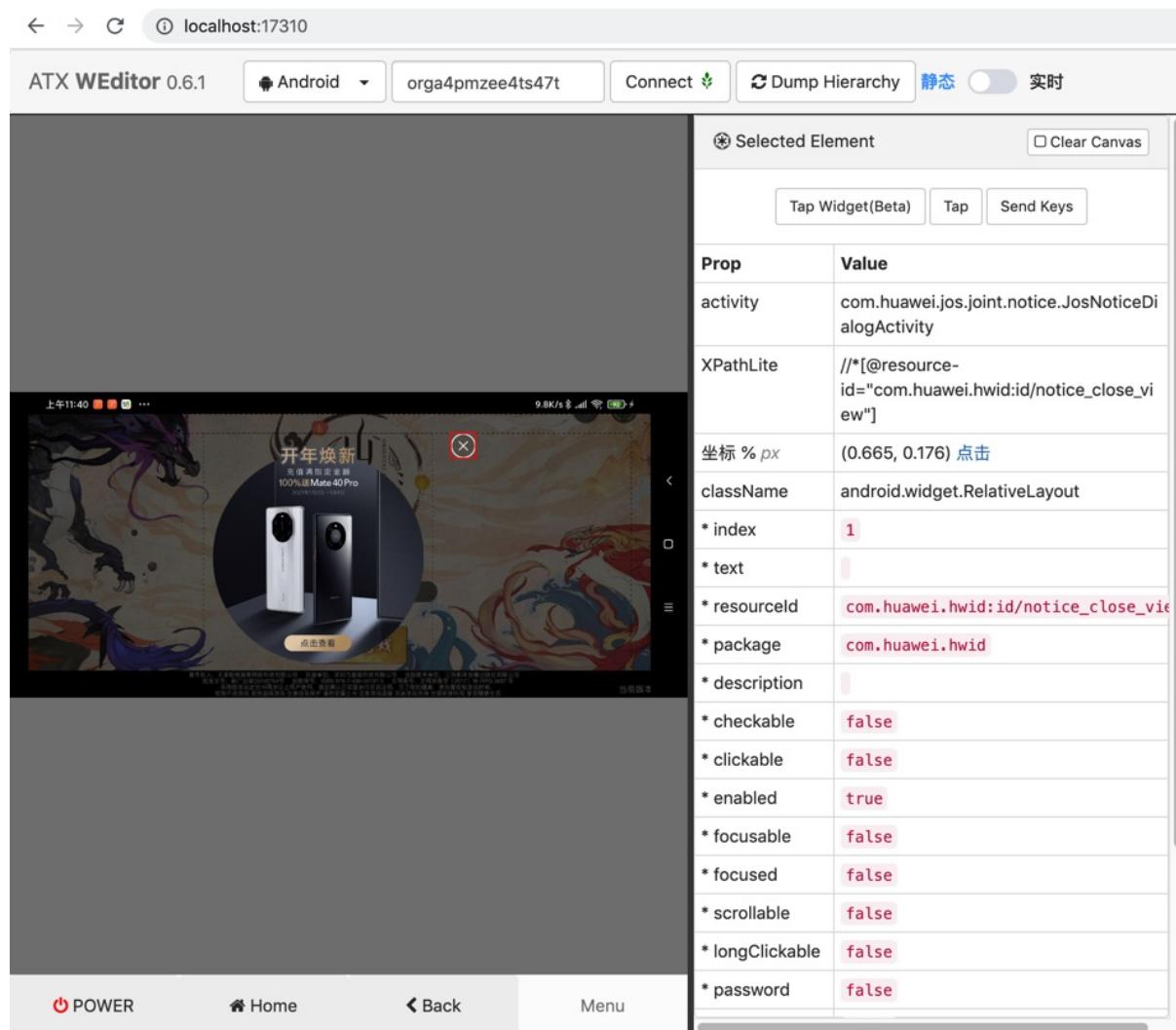
```

可以自动点击，很多游戏在注册和登录期间遇到的，在各大安卓应用市场同意授权返回后的广告类的弹框：

**华为应用市场 弹框广告**

**广告1**

weditor截图：



属性：

```

Prop      Value
activity   com.huawei.jos.joint.notice.JosNoticeDialogActivity
XPathLite  //*[@resource-id="com.huawei.hwid:id/notice_close_view"]
坐标 % px (0.665, 0.176) 点击
className  android.widget.RelativeLayout
* index    1
* text
* resourceId com.huawei.hwid:id/notice_close_view
* package   com.huawei.hwid
* description
* checkable false
* clickable false
* enabled   true
* focusable false
* focused   false
* scrollable false
* longClickable false
* password  false

```

```
* selected    false
# rect      {"x":1563,"y":146,"width":88,"height":88}
代码 d(resourceId "com.huawei.hwid:id/notice_close_view")
```

广告弹框被自动点击关闭后，相关log输出是：

```
your input:[I 210104 11:53:14 watcher:255] XPath(hook:): ["//android.widget.RelativeLayout
[contains(@resource-id, 'id/notice_close_view')]"]]
```

之后即可看到原先app的内容了：



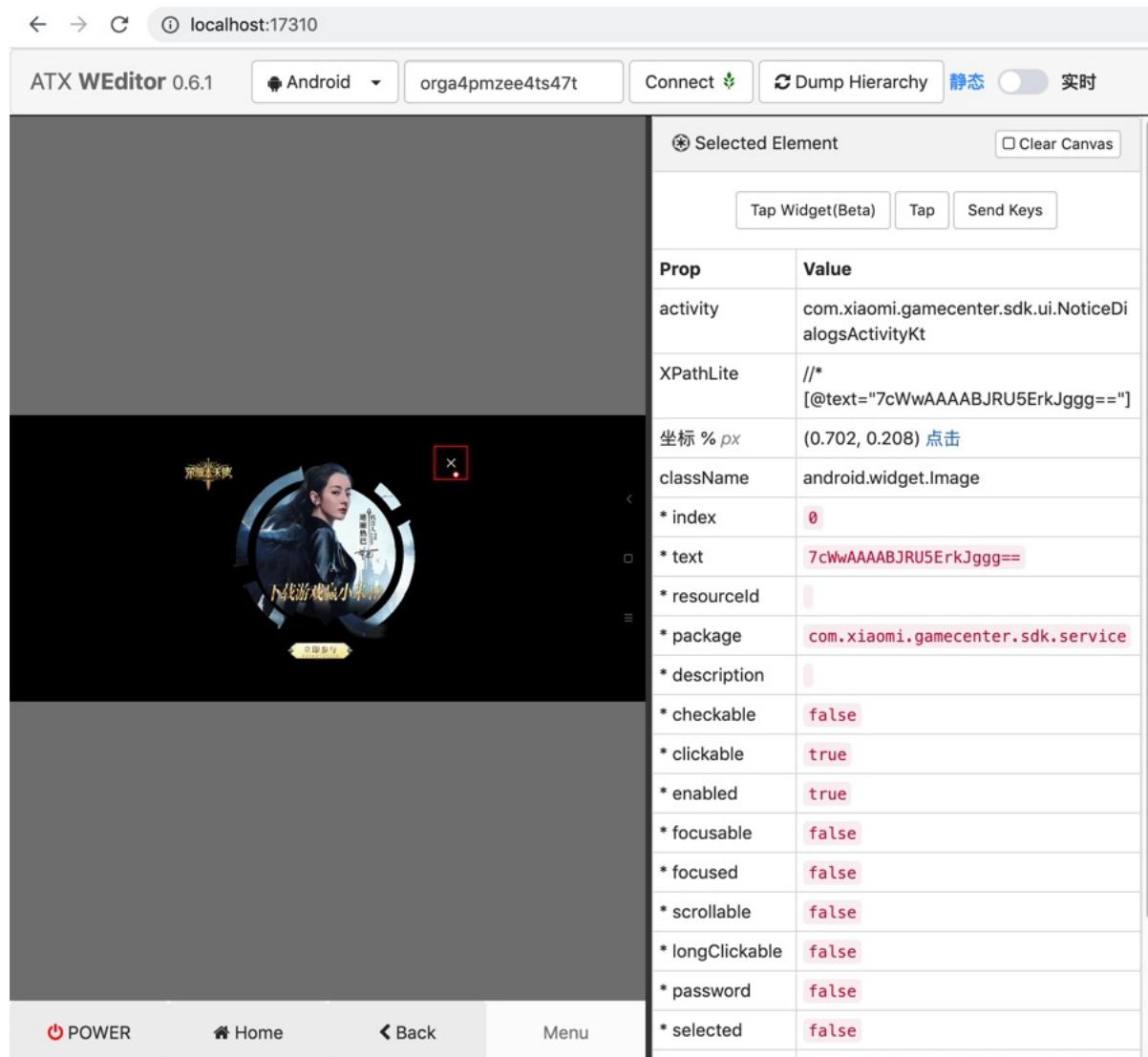
## 小米应用市场 弹框广告

### 广告1

游戏app截图：



weditor截图：



属性：

```

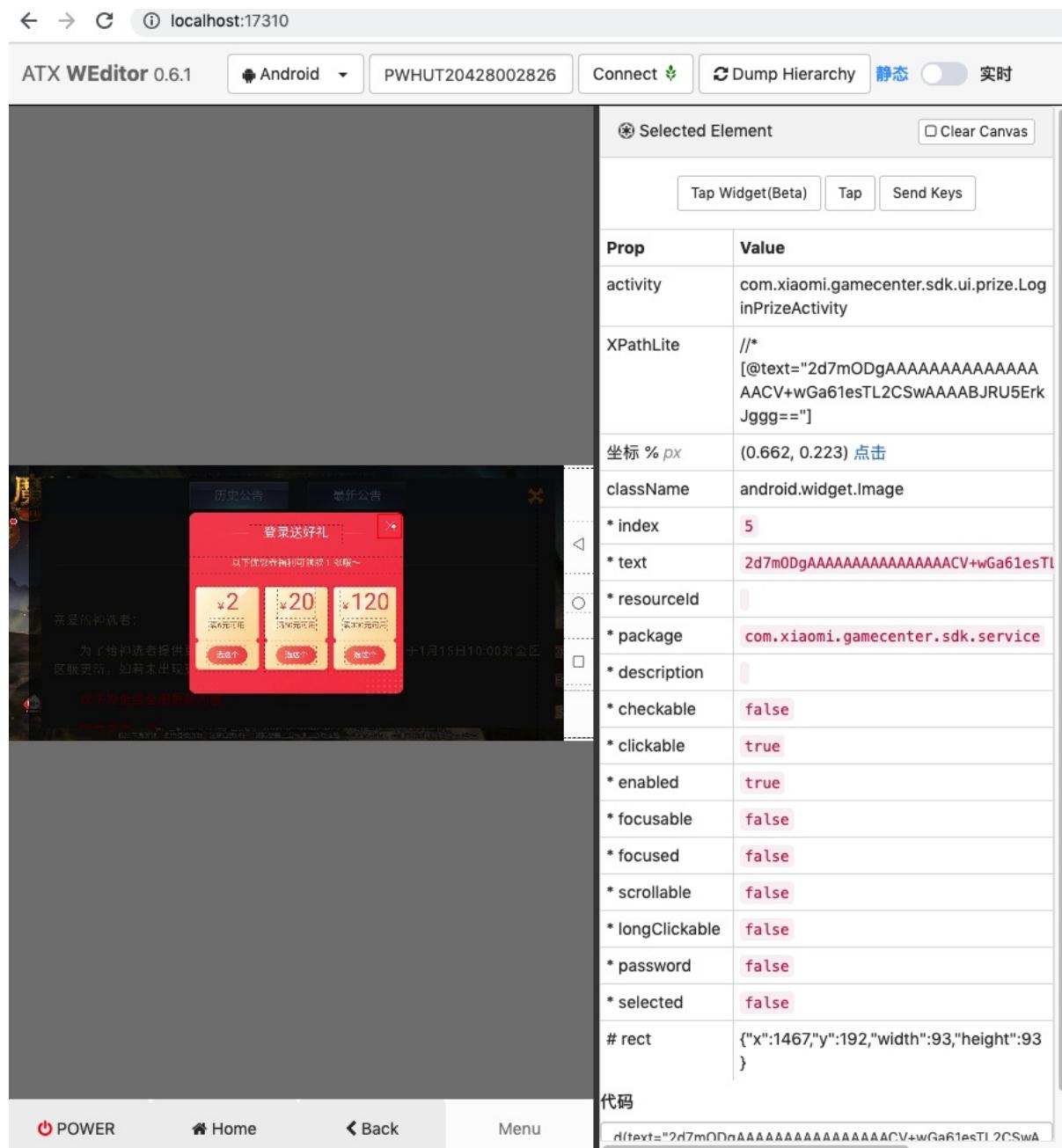
Prop      Value
activity   com.xiaomi.gamecenter.sdk.ui.NoticeDialogsActivityKt
XPathLite  //*[@text="7cWwAAAABJRU5ErkJgg=="]
坐标 % px  (0.702, 0.208) 点击
className  android.widget.Image
* index    0
* text     7cWwAAAABJRU5ErkJgg==
* resourceId
* package   com.xiaomi.gamecenter.sdk.service
* description
* checkable false
* clickable true
* enabled   true
* focusable false
* focused   false
* scrollable false
* longClickable false
* password   false
* selected   false

```

```
* selected false
# rect {"x":1603,"y":118,"width":124,"height":124}
代码 d(text "7cWwAAAABJRU5ErkJggg==")
```

## 广告2

weditor截图：



属性：

| Prop      | Value   |
|-----------|---|
| activity  | com.xiaomi.gamecenter.sdk.ui.prize.LoginPrizeActivity               |
| XPathLite | //*[@text="2d7mODgAAAAAAAAAAAAACV+wGa61esTL2CSwAAAABJRU5ErkJggg=="] |
| 坐标 % px   | (0.662, 0.223) 点击   |

```
className    android.widget.Image
* index      5
* text      2d7mODgAAAAAAAAAAAAACV+wGa61esTL2CSwAAAABJRU5ErkJggg==
* resourceId
* package    com.xiaomi.gamecenter.sdk.service
* description
* checkable   false
* clickable   true
* enabled     true
* focusable   false
* focused     false
* scrollable  false
* longClickable  false
* password    false
* selected    false
# rect      {"x":1467,"y":192,"width":93,"height":93}
代码 d(text "2d7mODgAAAAAAAAAAAAACV+wGa61esTL2CSwAAAABJRU5ErkJggg==")
```

## 广告3

游戏: com.yzcm.jr.mi/巨刃

游戏app截图:

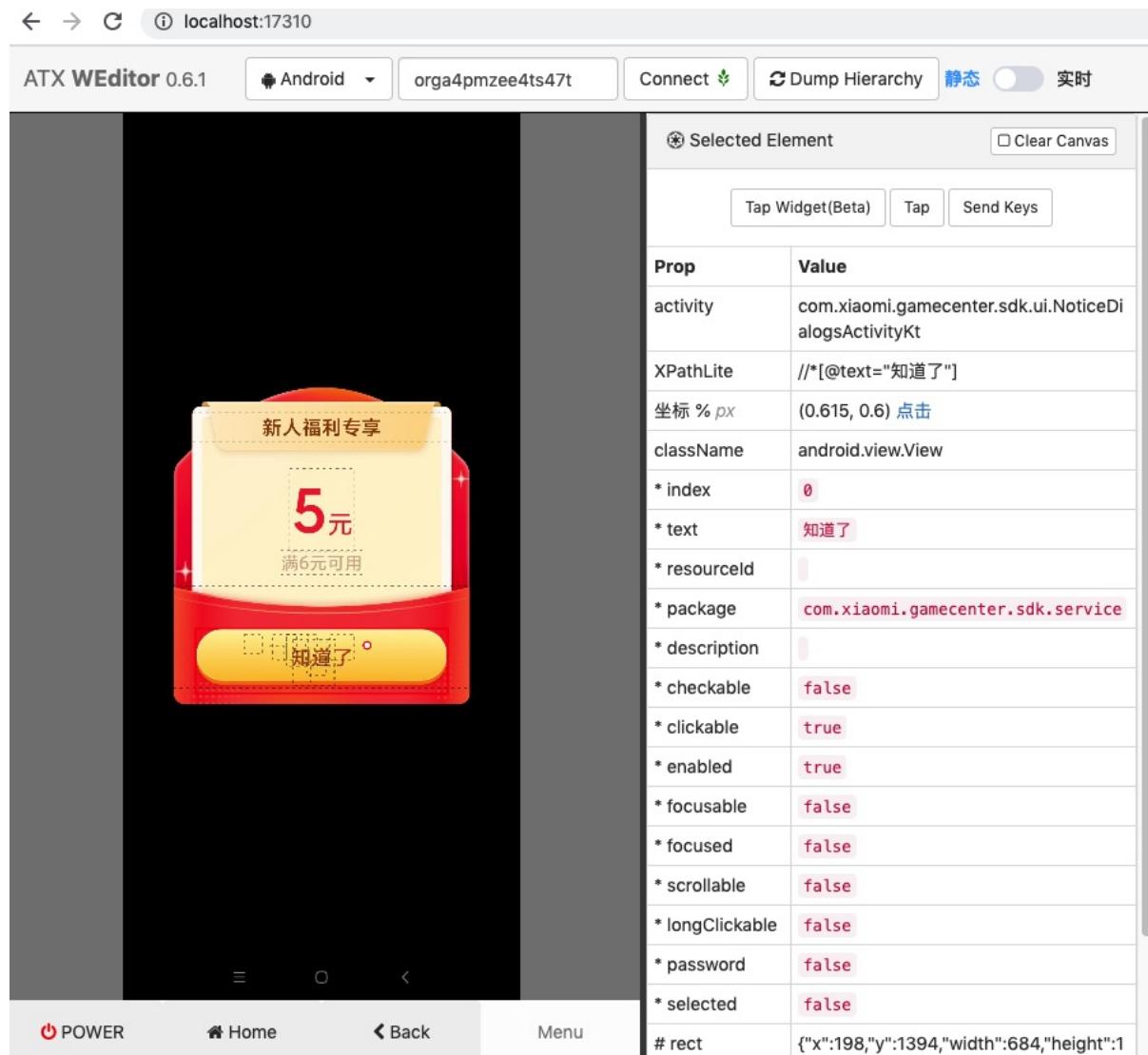


三

口

<

weditor截图：



属性：

```

Prop      Value
activity   com.xiaomi.gamecenter.sdk.ui.NoticeDialogsActivityKt
XPathLite  //*[@text="知道了"]
坐标 % px (0.615, 0.6) 点击
className  android.view.View
* index    0
* text     知道了
* resourceId
* package   com.xiaomi.gamecenter.sdk.service
* description
* checkable false
* clickable true
* enabled   true
* focusable false
* focused   false
* scrollable false

```

```
* longClickable  false
* password    false
* selected    false
# rect   {"x":198,"y":1394,"width":684,"height":176}
代码 d(text "知道了")
```

## 防止其他页面误判为小米广告

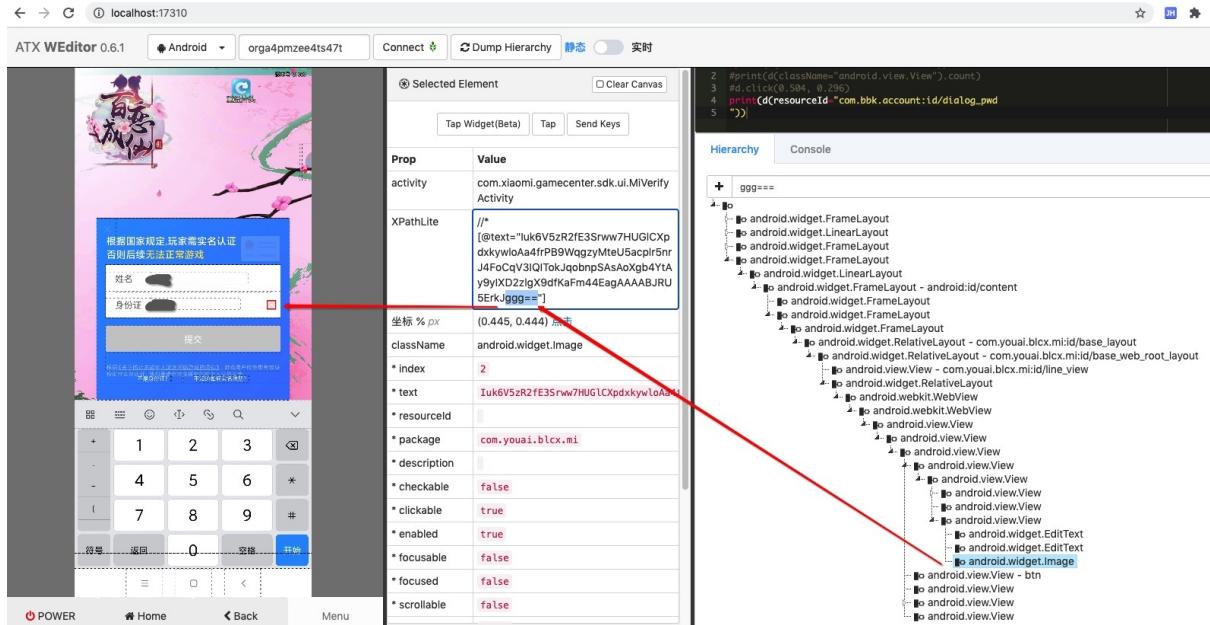
之前还遇到游戏： com.youai.blcx.mi/百恋成仙

会出现：

游戏app截图：



weditor截图：



会发现，当输入框后面的x关闭小按钮时，对应也有类似的字段：

```

Prop      Value
activity   com.xiaomi.gamecenter.sdk.ui.MiVerifyActivity
XPathLite  //*[@text="Iuk6V5zR2fE3Srw7HUG1CXpdxkywloAa4frPB9WqgzyMteU5acplr5nrJ4FoCqV3IQITokJqobnpSAsAoXgb4YtAy9yIXD2zlgX9dfKaFm44EagAAAABJRU5ErkJgg=="]
坐标 % px  (0.445, 0.444) 点击
className  android.widget.Image
* index    2
* text     Iuk6V5zR2fE3Srw7HUG1CXpdxkywloAa4frPB9WqgzyMteU5acplr5nrJ4FoCqV3IQITokJqobnpSAsAoXgb4YtAy9yIXD2zlgX9dfKaFm44EagAAAABJRU5ErkJgg==
* resourceId
* package   com.youai.blcx.mi
* description
* checkable false
* clickable true
* enabled   true
* focusable false
* focused   false
* scrollable false
* longClickable false
* password   false
* selected   false
# rect      {"x":878,"y":1056,"width":38,"height":38}

```

经过调试和思考，最后改为： wAAAABJRU5ErkJgg==

即：

```

# "//android.widget.Image[@text='7cWwAAAABJRU5ErkJgg==' and @index='0' and @clickable='true']", # 小米市场登录后 广告 弹框 关闭按钮 d(text="7cWwAAAABJRU5ErkJgg==")
# 小米市场登录后 广告 弹框 关闭按钮
# 1. d(text="7cWwAAAABJRU5ErkJgg==")

```

```
# 2. d(text="2d7mODgAAAAAAAAAAAAACV+wGa61esTL2CSwAAAABJRU5ErkJggg==")
# "//android.widget.Image[contains(@text, 'ggg==') and @clickable='true']",
# 防止误判 实名认证期间输入身份证时的清楚关闭按钮
# XPathLite    //*[@text="Iuk6V5zR2fE3Srww7HUG1CXpdxywloAa4frPB9WqgzyMteU5acplr5nrJ4FoC
qV3IQITokJqobnpSAsAoXgb4YtAy9yIXD2z1gX9dfKaFm44EagAAAABJRU5ErkJggg=="]
# 改为:
//android.widget.Image[contains(@text, 'wAAAABJRU5ErkJggg==') and @clickable='true'],"
```

即可防止误判：不要误判其他页面为小米的广告，而去关闭弹框了。

细节详见：

【已解决】用uiautomator2自动点击关闭小米应用市场登录后的广告弹框

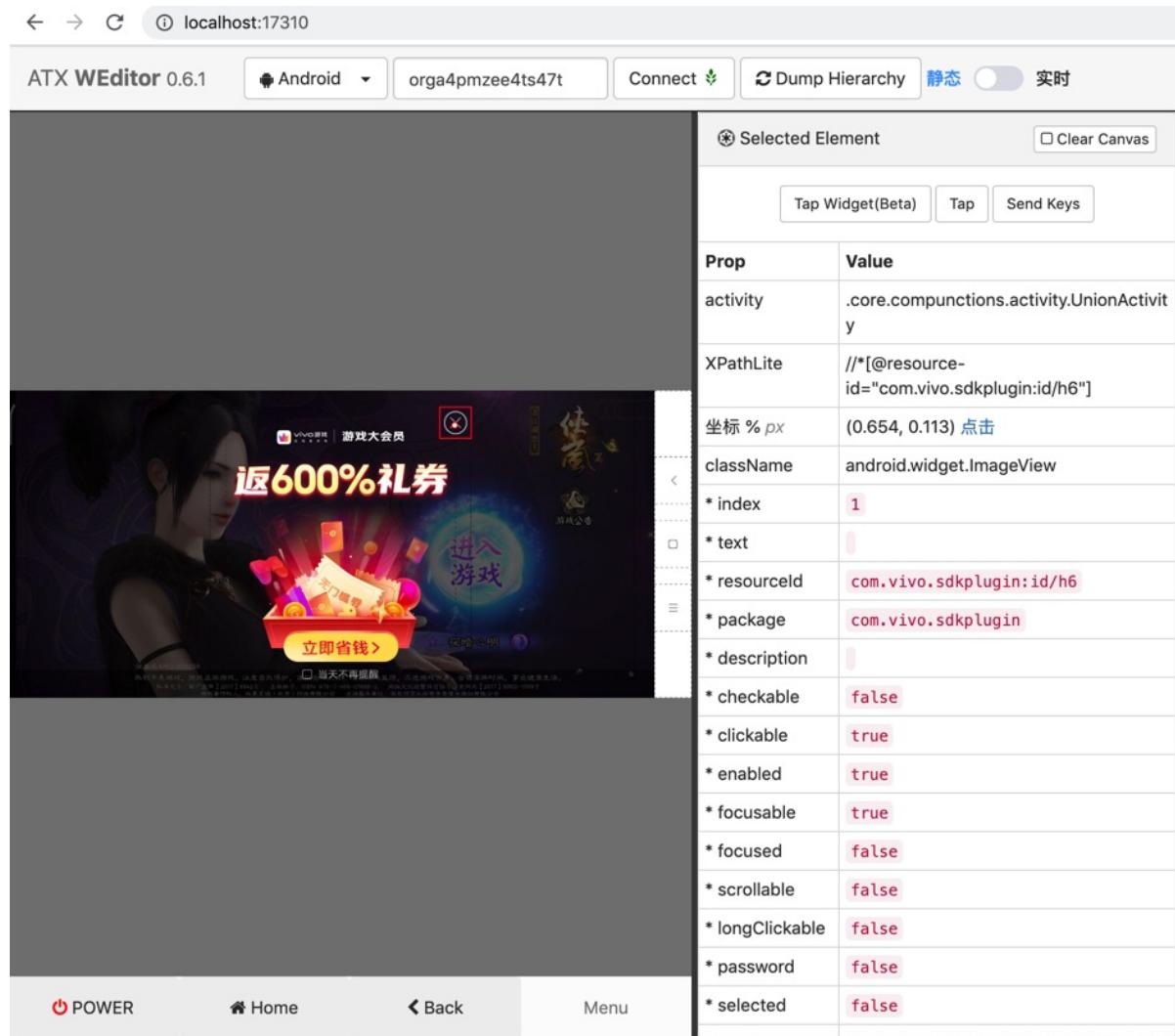
## Vivo应用市场 弹框广告

### 广告1

游戏app截图：



weditor截图：



属性：

```

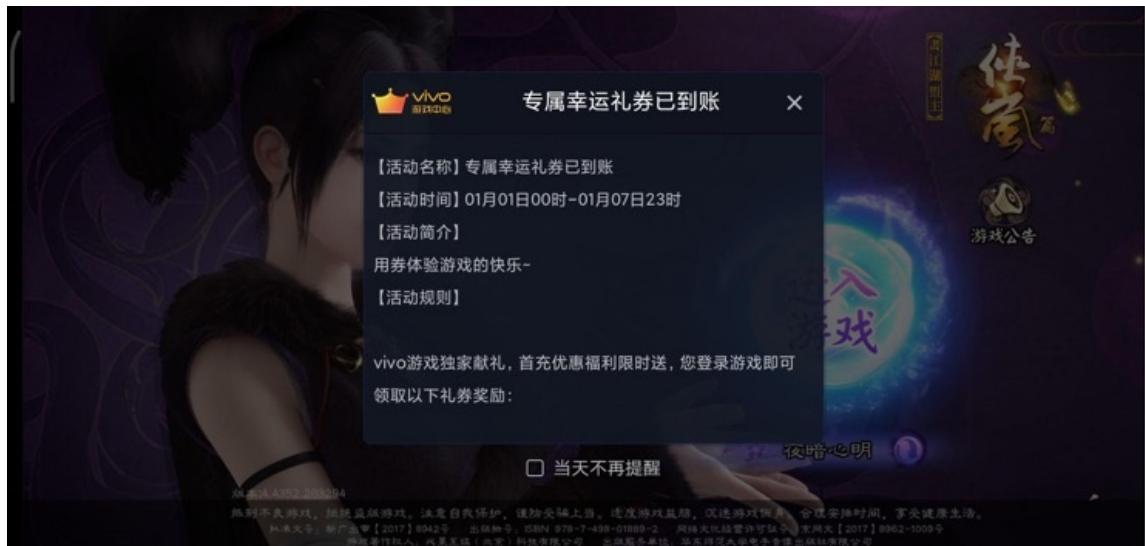
Prop      Value
activity   .core.compunctions.activity.UnionActivity
XPathLite  //*[@resource-id="com.vivo.sdkplugin:id/h6"]
坐标 % px (0.654, 0.113) 点击
className  android.widget.ImageView
* index    1
* text
* resourceId com.vivo.sdkplugin:id/h6
* package    com.vivo.sdkplugin
* description
* checkable  false
* clickable  true
* enabled    true
* focusable  true
* focused    false
* scrollable false
* longClickable false
* password   false
* selected   false
# rect      {"x":1517,"y":58,"width":110,"height":110}

```

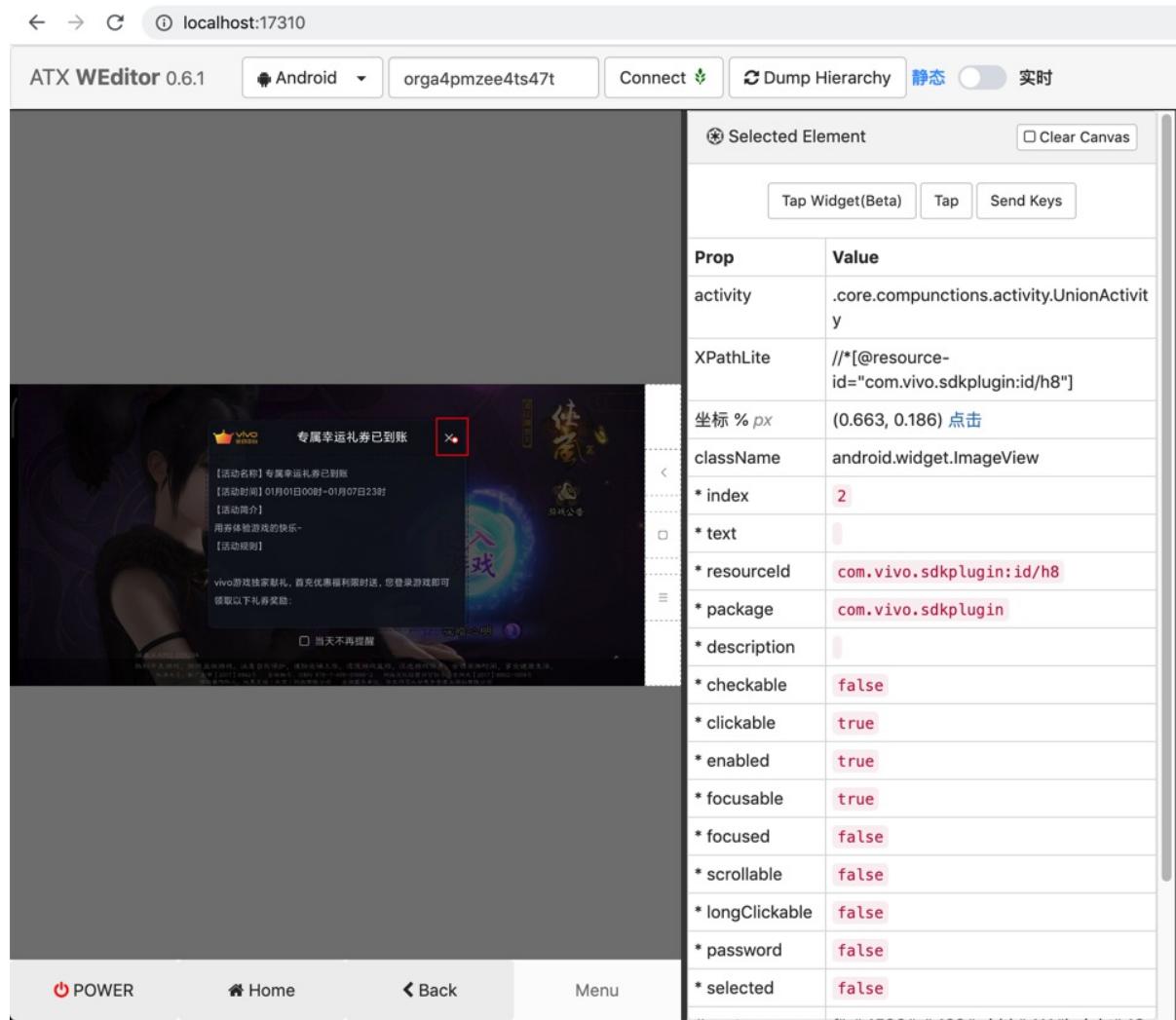
代码 d(resourceId="com.vivo.sdkplugin:id/h6")

## 广告2

游戏app截图：



weditor截图：



属性:

```

Prop      Value
activity   .core.compunctions.activity.UnionActivity
XPathLite  //*[@resource-id="com.vivo.sdkplugin:id/h8"]
坐标 % px  (0.663, 0.186) 点击
className  android.widget.ImageView
* index    2
* text
* resourceId com.vivo.sdkplugin:id/h8
* package   com.vivo.sdkplugin
* description
* checkable false
* clickable true
* enabled   true
* focusable true
* focused   false
* scrollable false
* longClickable false
* password  false
* selected  false
# rect     {"x":1526,"y":122,"width":111,"height":132}

```

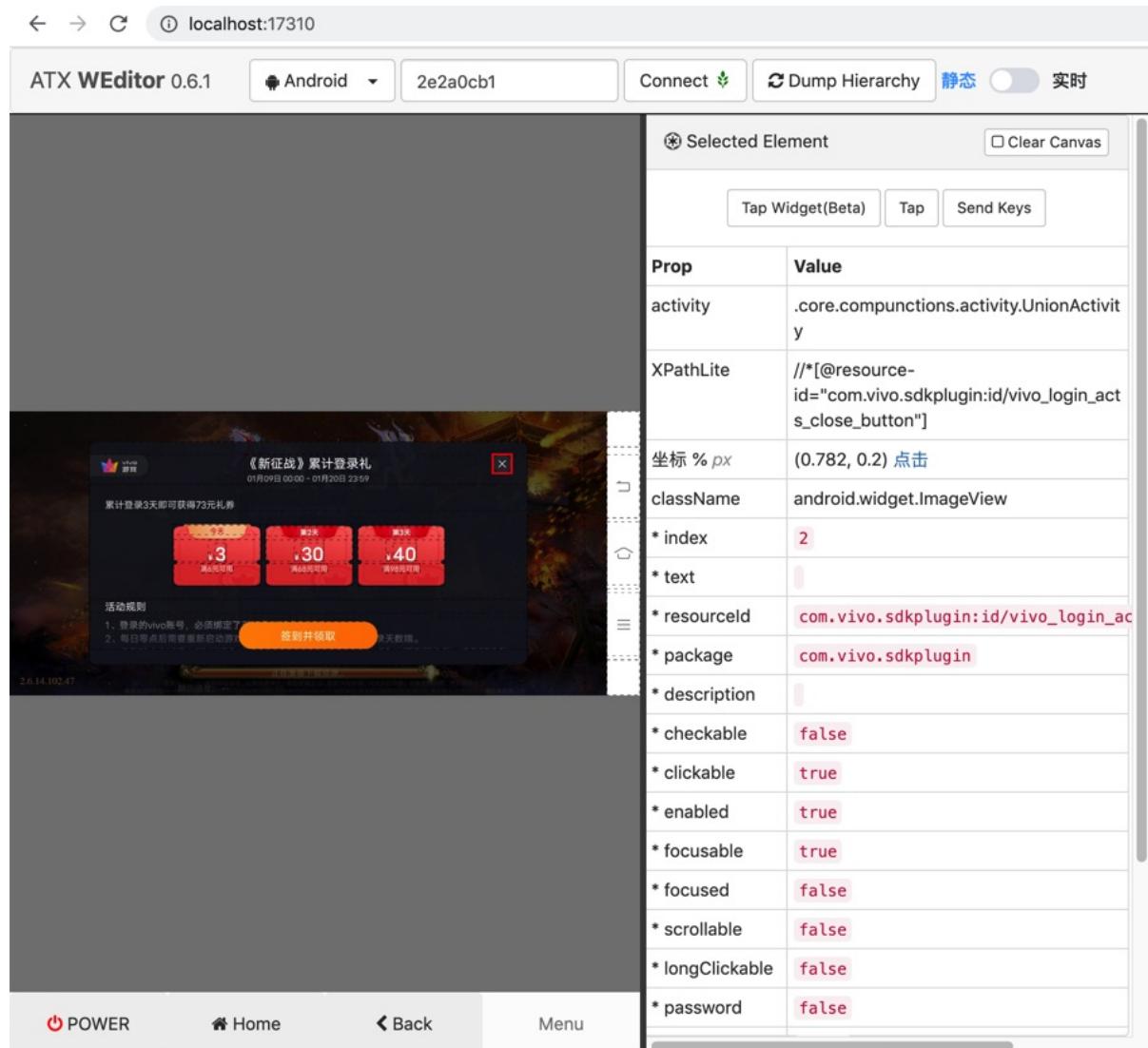
代码 `d(resourceId="com.vivo.sdkplugin:id/h8")`

## 广告3

游戏app截图：



weditor截图：



属性：

```

Prop      Value
activity   .core.compunctions.activity.UnionActivity
XPathLite  //*[@resource-id="com.vivo.sdkplugin:id/vivo_login_acs_close_button"]
坐标 % px (0.782, 0.2) 点击
className  android.widget.ImageView
* index    2
* text
* resourceId com.vivo.sdkplugin:id/vivo_login_acs_close_button
* package   com.vivo.sdkplugin
* description
* checkable false
* clickable true
* enabled   true
* focusable true
* focused   false
* scrollable false
* longClickable false
* password  false
* selected  false

```

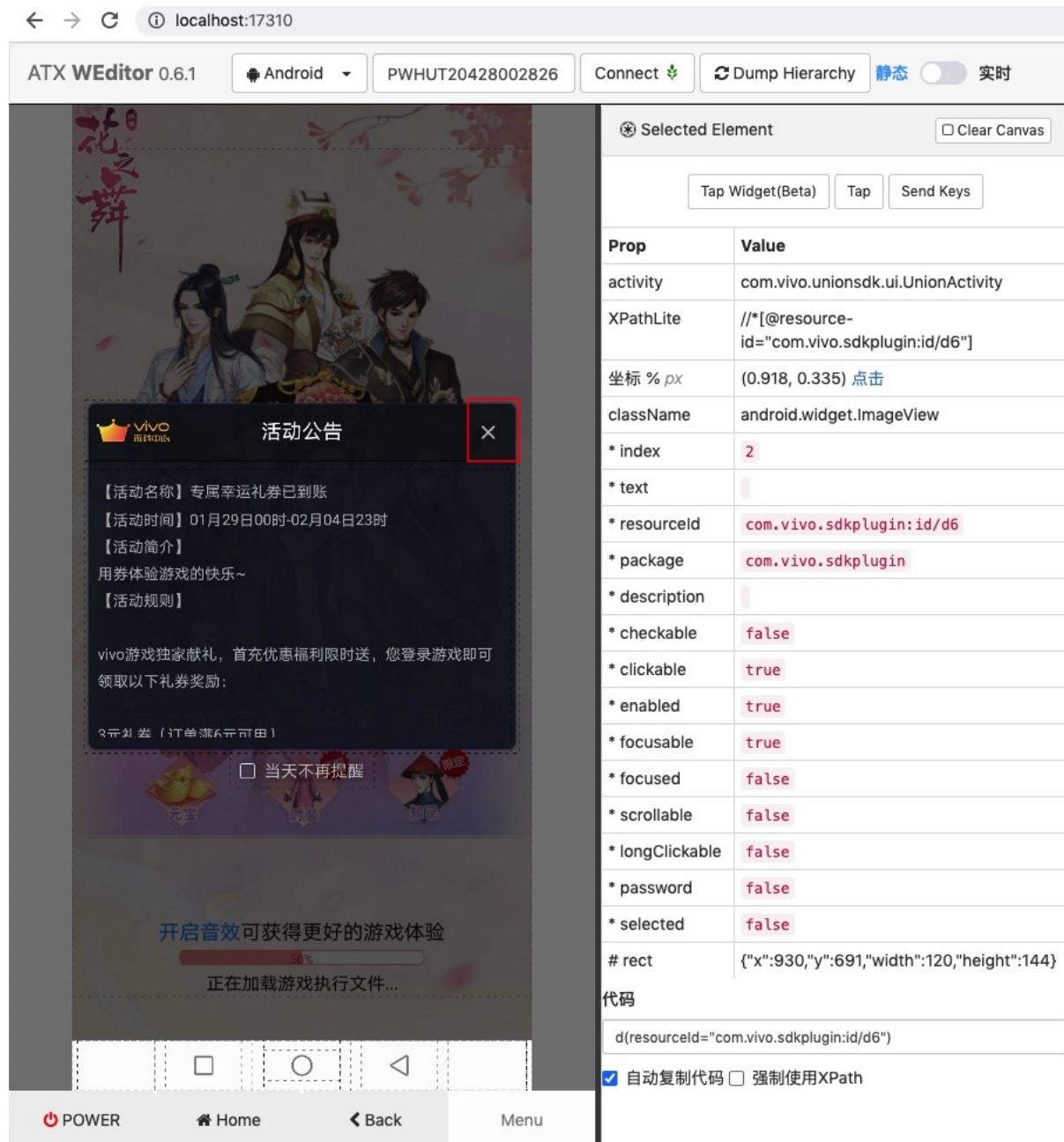
```
# rect {"x":1226,"y":109,"width":48,"height":48}  
代码 d(resourceId="com.vivo.sdkplugin:id/vivo_login_act_close_button")
```

## 广告4

游戏app截图：



weditor截图：



核心属性：

- resourceId : com.vivo.sdkplugin:id/d6

crifan.com, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新：2021-07-18 09:59:23

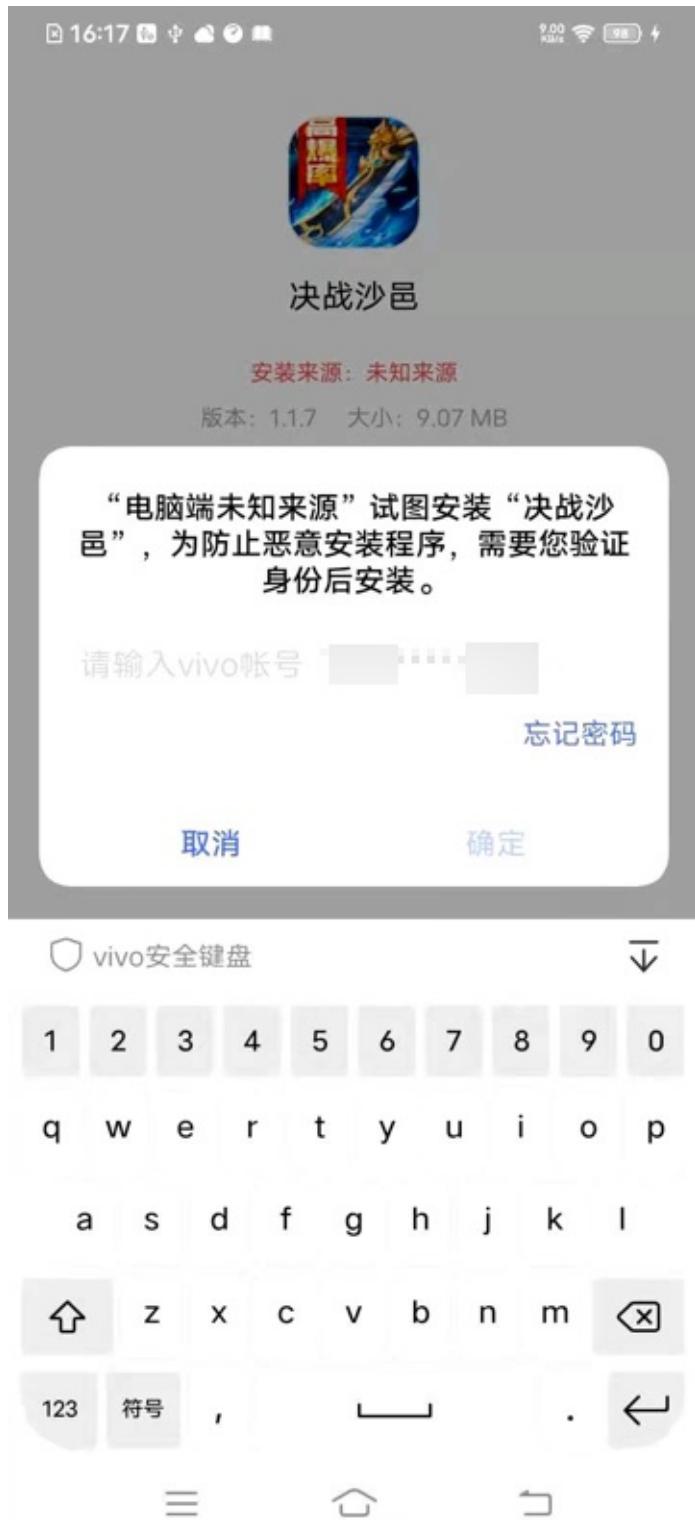
## Vivo手机中安装应用时，自动输入账号密码

背景：

用 adb 去给安卓手机安卓 apk ， 比如：

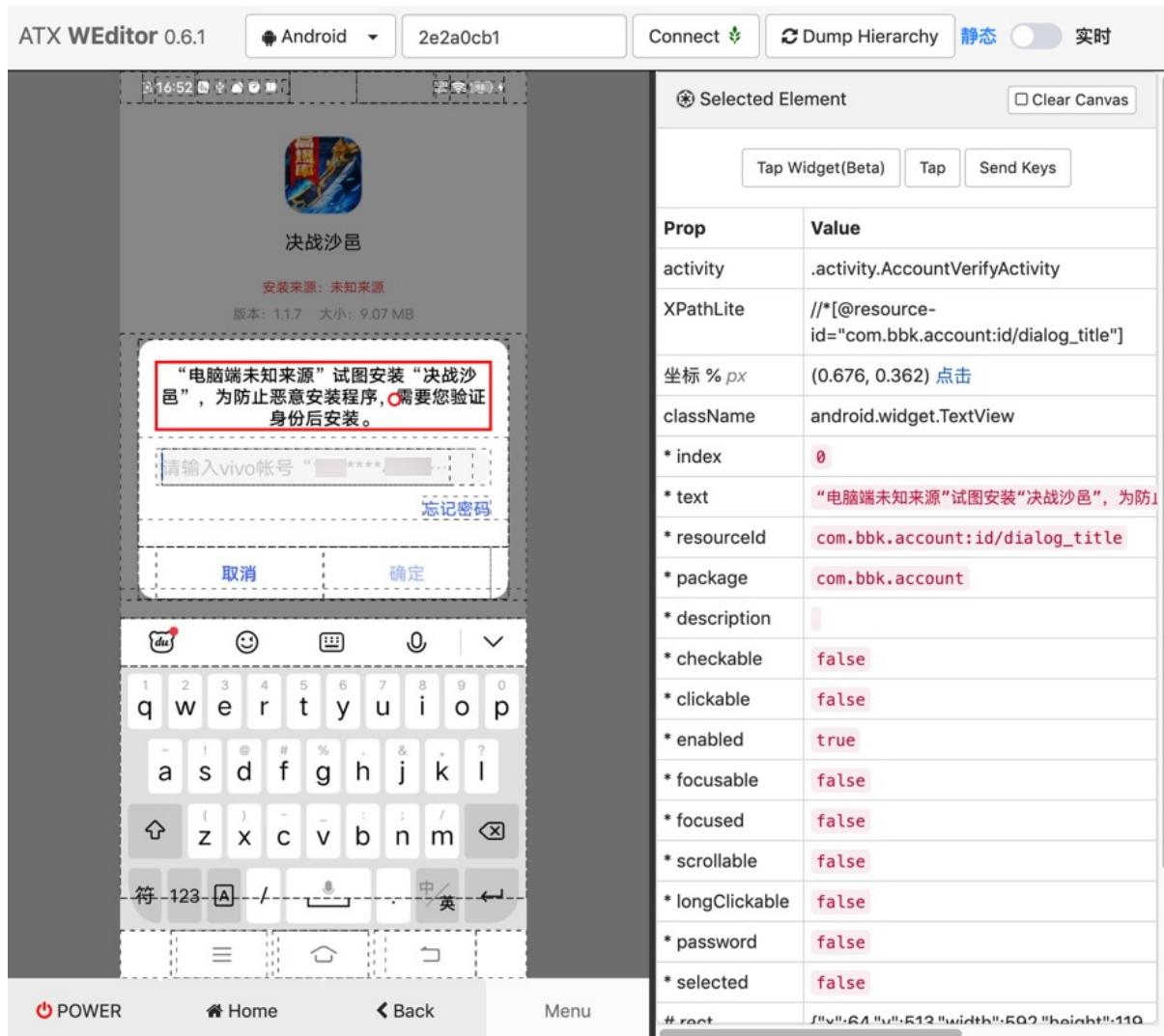
```
adb -s 2e2a0cb1 install 决战沙邑.apk
```

时，Vivo手机，由于默认安全限制很死，导致无法关闭安全验证，会弹框：



必须手动输入账号和密码后才能继续安装。

对应的weditor截图：



属性是：

```

Prop      Value
activity   .activity.AccountVerifyActivity
XPathLite  //*[@resource-id="com.bbk.account:id/dialog_title"]
坐标 % px (0.676, 0.362) 点击
className  android.widget.TextView
* index    0
* text     “电脑端未知来源”试图安装“决战沙邑”，为防止恶意安装程序，需要您验证身份后安装。
* resourceId com.bbk.account:id/dialog_title
* package   com.bbk.account
* description
* checkable false
* clickable false
* enabled   true
* focusable false
* focused   false
* scrollable false
* longClickable false
* password  false
* selected  false

```

```
# rect {"x":64,"y":513,"width":592,"height":119}
代码 d(resourceId="com.bbk.account:id/dialog_title")
```

经过调试，最终实现了自动化输入账号和密码，得以自动继续安装。

代码：

```
Vivo_Account: "yourPhone" # Vivo account
Vivo_Password: "yourPassword" # password for Vivo account

Vivo_Password_Input_Xpath: "//android.widget.LinearLayout[@resource-id='com.bbk.account:id/dialog_pwd']"

self.driver.watcher.when(Vivo_Password_Input_Xpath]).call(self.autoInputVivoPassword)

def selectorSetText(self, curXpathSelector, inputText):
    # Special: add click to try workaround for 360 pwd EditText input but input to 360 account EditText
    # curXpathSelector.click()
    # curXpathSelector.clear_text()
    selectorSetTextResp = curXpathSelector.set_text(inputText)
    logging.debug("selectorSetTextResp=%s", selectorSetTextResp) # selectorSetTextResp=None

    # 在set_text后，输入法会变成FastInputIME输入法
    # 用下面代码可以实现：关掉FastInputIME输入法，切换回系统默认输入法
    self.driver.set_fastinput_ime(False)

def autoInputVivoPassword(self):
    """Auto input Vivo account password"""
    logging.info("Try auto input vivo password")

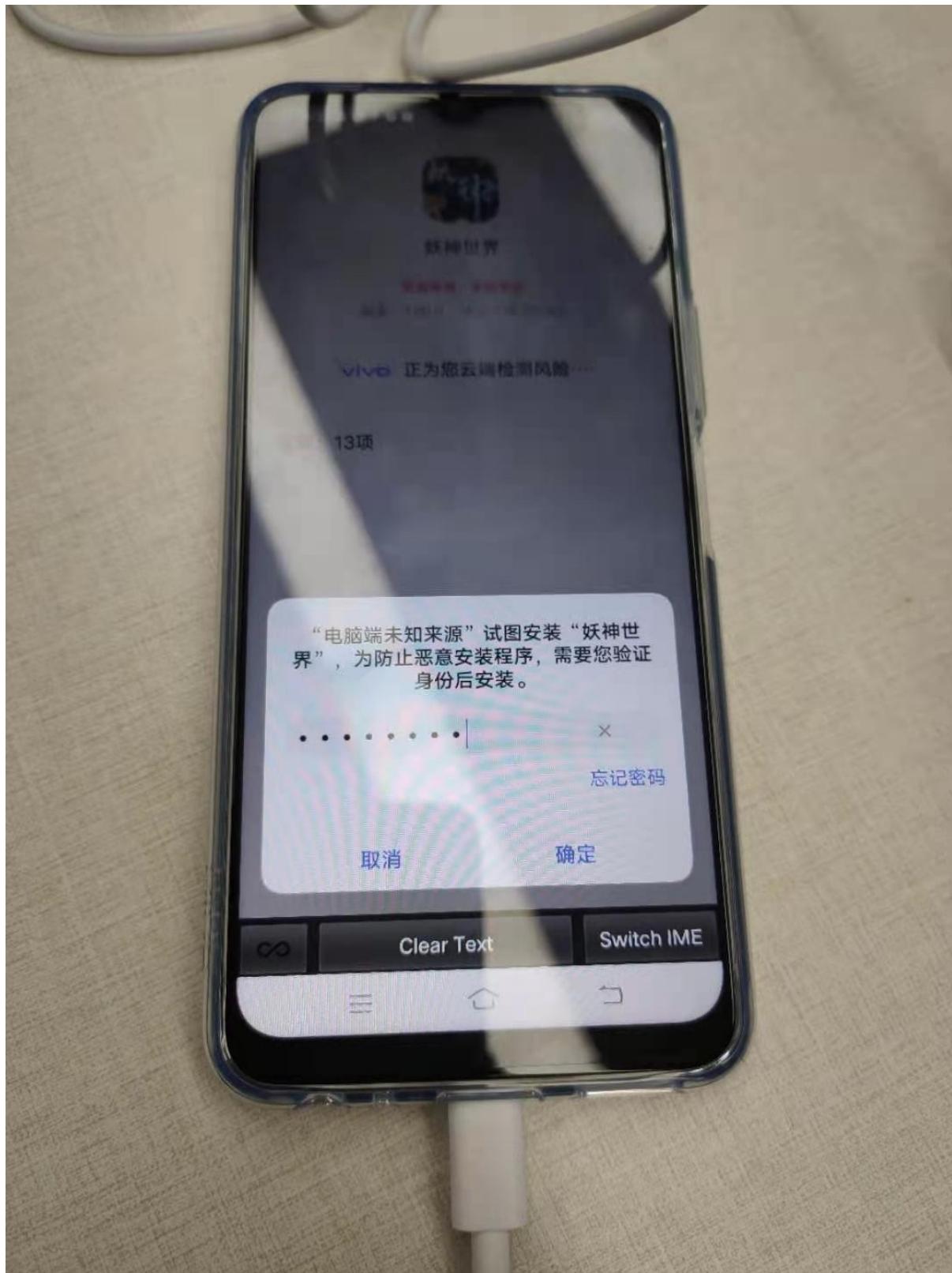
    pwdDiaglogSelector = self.driver.xpath(Vivo_Password_Input_Xpath)
    logging.debug("pwdDiaglogSelector=%s", pwdDiaglogSelector)
    # PwdDiaglogSelector=XPathSelector("//android.widget.LinearLayout[@resource-id='com.bbk.account:id/dialog_pwd']")
    logging.info("%s to found password dialog", pwdDiaglogSelector.exists)
    # selectorSetTextResp = pwdDiaglogSelector.set_text(Vivo_Password)
    # logging.debug("selectorSetTextResp=%s", selectorSetTextResp)
    # # selectorSetTextResp=None
    # # 在set_text后，输入法会变成FastInputIME输入法
    # # 用下面代码可以实现：关掉FastInputIME输入法，切换回系统默认输入法
    # self.driver.set_fastinput_ime(False)
    self.selectorSetText(pwdDiaglogSelector, Vivo_Password)
    logging.info("Has input password to dialog")

    okButtonText = "确定"
    okButtonElement = self.driver(text(okButtonText, className="android.widget.Button", resourceId="android:id/button1"))
    logging.debug("okButtonElement=%s", okButtonElement)
    logging.info("%s to found %s button", okButtonElement.exists, okButtonText)
    if okButtonElement.exists:
        okButtonElement.click()
        logging.info("Clicked %s for vivo password")
```

```
logging.info("Complete auto input vivo password")
```

对应自动化操作期间的手机截图：

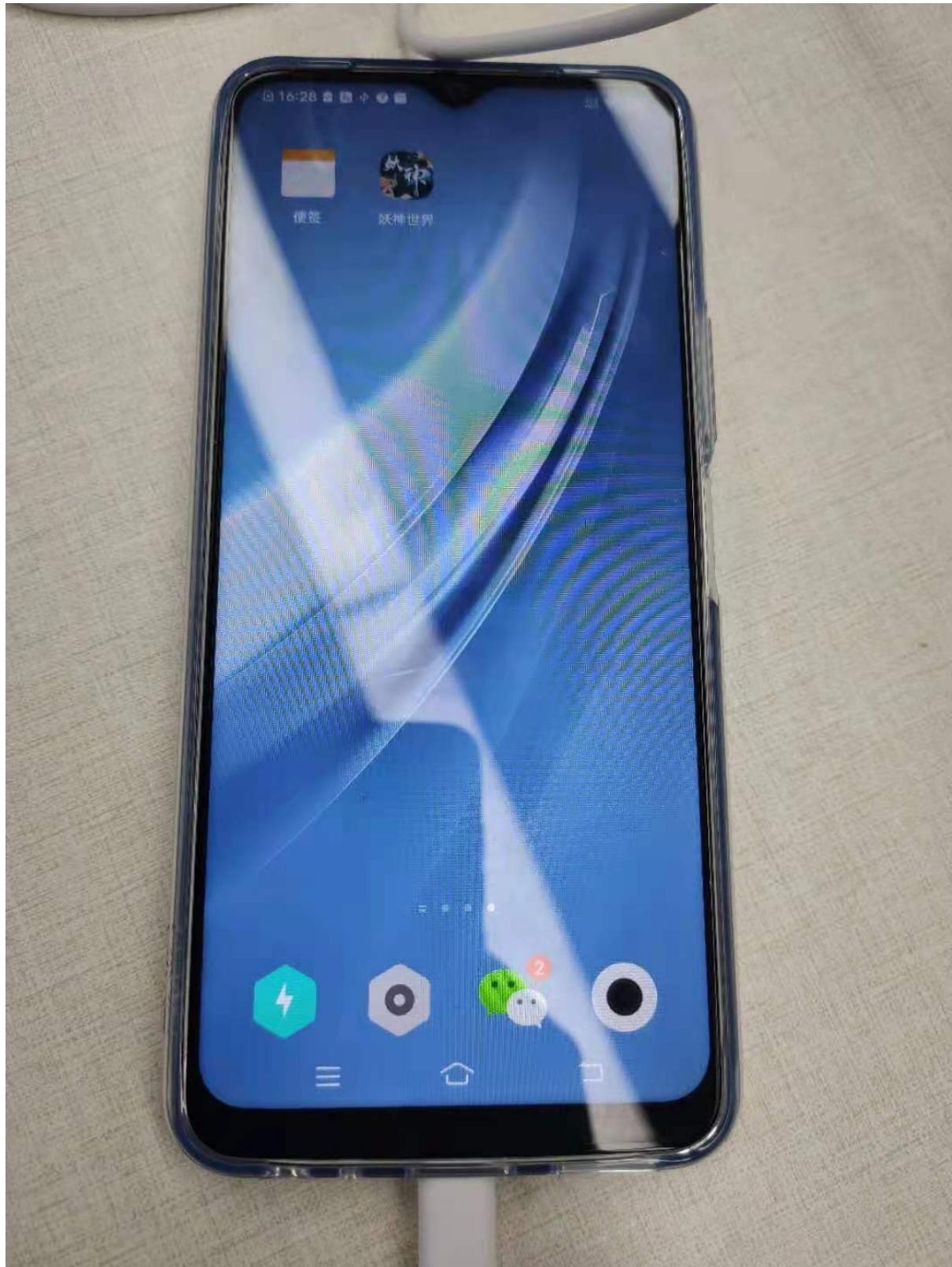
自动输入了密码：



点击了 确定后，再点击点击 继续安装



之后即可在手机中看到成功安装的app:



相关shell终端中的日志是：

```
adb -s 2e2a0cb1 install 斩月屠龙_13.5MB.apk
Performing Streamed Install
Success
```

整个调试过程的log是：

```
_VivoGameXianXiaChuanQi4th10_gameApp_Android_1.apk
Performing Streamed Install
[I 201216 09:34:37] watcher:255] XPath(hook:): ["//android.widget.LinearLayout[@resource-id='com.bbk.account:id/dialog_pwd']"]
[201216 09:34:37][DevicesMethods.py 245] Try auto input vivo password
[201216 09:34:37][DevicesMethods.py 250] True to found password dialog
[201216 09:34:39][DevicesMethods.py 254] Has input password to dialog
[201216 09:34:39][DevicesMethods.py 263] True to found 确定 button
[201216 09:34:40][DevicesMethods.py 266] Clicked 确定 for vivo password
[I 201216 09:34:46] watcher:255] XPath(hook:继续安装): ['继续安装']
Success
```

```
终端 问题 4 输出 调试控制台 4: Python Debug Console + × × ×
[201216 09:34:13] [DevicesMethods.py 4122] start to install app in 20201216_fengyun_VivoGameXianXiaChuanQi4th10_gameApp_Android.txt
[201216 09:34:13] [DevicesMethods.py 4162] file size: 388.3MB
[201216 09:34:13] [DevicesMethods.py 4178] Run length=265, command: adb -s 2e2a0cb1 install -r -f /Users/Limao/dev/crawler/appAutoCrawler/AppCrawler/task/20201216_fengyun_VivoGameXianXiaChuanQi4th10/20201216_fengyun_VivoGameXianXiaChuanQi4th10_gameApp_Android/20201216_fengyun_VivoGameXianXiaChuanQi4th10_gameApp_Android_1.apk
Performing Streamed Install
[I 201216 09:34:37] watcher:255] XPath(hook:): ["//android.widget.LinearLayout[@resource-id='com.bbk.account:id/dialog_pwd']"]
[201216 09:34:37][DevicesMethods.py 245] Try auto input vivo password
[201216 09:34:37][DevicesMethods.py 250] True to found password dialog
[201216 09:34:39][DevicesMethods.py 254] Has input password to dialog
[201216 09:34:39][DevicesMethods.py 263] True to found 确定 button
[201216 09:34:40][DevicesMethods.py 266] Clicked 确定 for vivo password
[I 201216 09:34:46] watcher:255] XPath(hook:继续安装): ['继续安装']
Success
[201216 09:34:51] [AppCrawler.py 137] crawl com.p2y9y.quxuan.xianjianol.vivo/修仙OL start
[W 201216 09:34:51] watcher:181 already started
[201216 09:34:51] [DevicesMethods.py 181] start com.p2y9y.quxuan.xianjianol.vivo
```

详见：

[【已解决】用Python的uiautomator2自动识别和输入vivo账号密码以自动安装安卓apk](#)

[【已解决】uiautomator2中如何自动实现检测发现匹配元素就执行对应回调函数](#)

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2021-07-18 09:59:23

## Vivo账号自动登录

自动化输入vivo账号和密码并登录

背景：

测试游戏期间，遇到非Vivo手机，需要手动输入Vivo的账号和密码，才能继续测试。

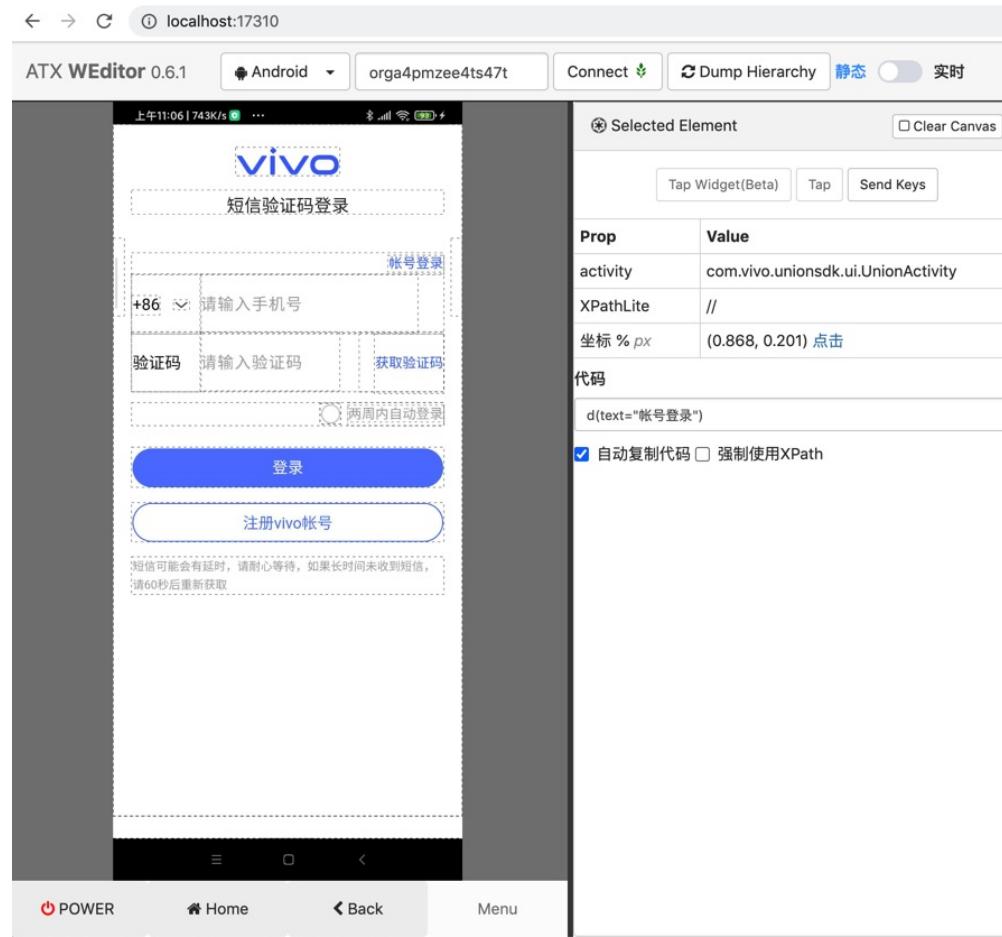
所以写代码将此过程自动化。

具体过程：

- 短信验证码登录页

◦

- weditor调试



- 点击：账号登录 按钮
  - 切换页面到：账号登录页
  - 支持已经切换到 账号登录页，而不会误点击 中间顶部的 账号登录 文字
- 账号登录页

- ◦ 请输入手机号：输入手机号



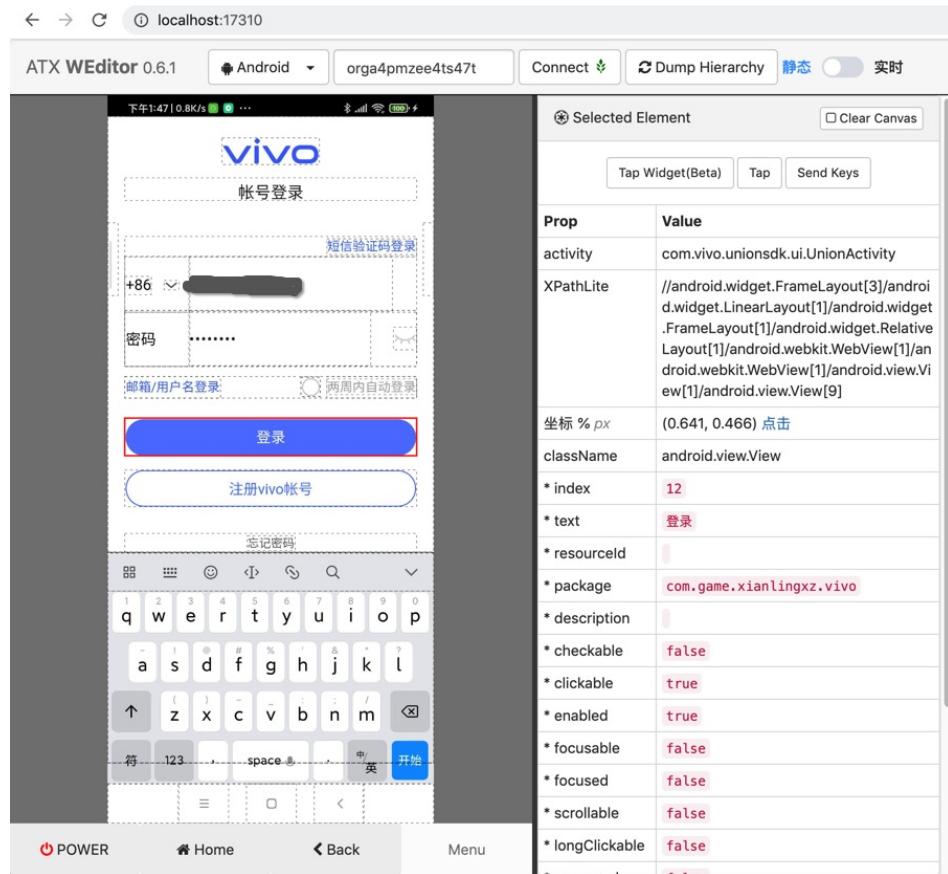
- 支持已经输入手机号，再次重新输入手机号
- 此处输入手机号会触发输入法切换到 fastIME
- 请输入密码：输入密码



■ 真机效果



■ weditor调试



- 支持已经输入密码，再次重新输入密码
- 点击：登录 按钮

代码：

原始独立测试代码：

```

import time
import uiautomator2 as u2

def selectorSetText(u2Dev, curXPathSelector, inputText):
    selectorSetTextResp = curXPathSelector.set_text(inputText)
    logging.info("selectorSetTextResp=%s", selectorSetTextResp) # selectorSetTextResp=None
    # 在set_text后，输入法会变成FastInputIME输入法
    # 用下面代码可以实现：关掉FastInputIME输入法，切换回系统默认输入法
    u2Dev.set_fastinput_ime(False)

#####
# for Redmi 10X: auto do vivo account login, input phone and password to login
#####

def autoDoVivoLogin(u2Dev):
    doScreenshot(u2Dev)

    RegVivoAccountStr = "注册vivo帐号"
    regVivoAccountXPath = """//android.view.View[@text="%s"]""" % RegVivoAccountStr
    # '//android.view.View[@text="注册vivo帐号"]'
    regVivoAccountSelector = u2Dev.xpath(regVivoAccountXPath)

```

```

if regVivoAccountSelector.exists:
    logging.info("Found %s", RegVivoAccountStr)
    accountLoginStr = "帐号登录"
    # accountLoginXpath = """/android.view.View[@text="%s"]"""\ % accountLoginStr
    # #
    # accountLoginSelector = u2Dev.xpath(accountLoginXpath)
    # if accountLoginSelector.exists:
    #     logging.info("Found %s", accountLoginStr)
    #     accountLoginSelector.click()
    # accountLoginElement = u2Dev(text=accountLoginStr, className="android.view.View")
    accountLoginElement = u2Dev(text=accountLoginStr, className="android.view.View", c
lickable="true")
    logging.info("accountLoginElement=%s", accountLoginElement) # accountLoginElement=
<uiautomator2._selector.UiObject object at 0x10dd4be80>
    logging.info("accountLoginElement.exists=%s", accountLoginElement.exists) # accoun
tLoginElement.exists=True
    if accountLoginElement.exists:
        accountLoginElement.click()
        logging.info("Has clicked %s button", accountLoginStr)

        time.sleep(0.1)
        doScreenshot(u2Dev)
    else:
        logging.warning("Not found %s button", accountLoginStr)

    # phoneXPath = """//android.widget.EditText[@text="" and @index="5"]"""
    phoneXPath = """//android.widget.EditText[@index="5" and @clickable="true" and @fo
cusable="true"]"""
    phoneSelector = u2Dev.xpath(phoneXPath)
    if phoneSelector.exists:
        logging.info("Found 请输入手机号")
        # phoneSelector.set_text(Vivo_Account)
        selectorSetText(u2Dev, phoneSelector, Vivo_Account)
        logging.info("Has input vivo account phone number")
    else:
        logging.warning("Not found 请输入手机号")

    passwordStr = "请输入密码"
    # passwordXPath = """//android.widget.EditText[@text="%s" and @index="2" and @cli
ckable="true"]"""\ % passwordStr
    passwordXPath = """//android.widget.EditText[@index="2" and @clickable="true" and
@focusable="true"]"""
    passwordSelector = u2Dev.xpath(passwordXPath)
    if passwordSelector.exists:
        logging.info("Found %s", passwordStr)
        # pwdClickResp = passwordSelector.click()
        # logging.debug("pwdClickResp=%s", pwdClickResp)
        # doScreenshot(u2Dev)
        selectorSetText(u2Dev, passwordSelector, Vivo_Password)
        logging.info("Has input vivo password")
    else:
        logging.warning("Not found %s", passwordStr)

```

```

loginStr = "登录"
# loginXPath = """//android.view.View[@text="登录" and @clickable="true"]"""
loginXPath = """//android.view.View[@text="登录" and @index="12" and @clickable="true"]"""
loginSelector = u2Dev.xpath(loginXPath)
if loginSelector.exists:
    loginSelector.click()
    logging.info("Has clicked %s button", loginStr)
    doScreenshot(u2Dev)
else:
    logging.warning("Not found %s", loginStr)

def androidAutomation():
    u2Dev = u2.connect(DeviveId)
    logging.info("u2Dev=%s", u2Dev) # u2Dev=<uiautomator2.Device object at 0x1079bee20>
    ...
    autoDoVivoLogin(u2Dev)

```

合并到项目后：

```

Vivo_Register_Vivo_Account_Xpath: "//android.view.View[@text='注册vivo帐号']"

self.driver.watcher.when(self.config["Vivo_Register_Vivo_Account_Xpath"]).call(self.autoDoVivoAccountLogin)

def autoDoVivoAccountLogin(self):
    """Auto do Vivo account login"""
    logging.info("Try auto do vivo account login")

    accountLoginStr = "帐号登录"
    # accountLoginXPath = """//android.view.View[@text="%s"]""" % accountLoginStr
    # #
    # accountLoginSelector = self.driver.xpath(accountLoginXPath)
    # if accountLoginSelector.exists:
    #     logging.info("Found %s", accountLoginStr)
    #     accountLoginSelector.click()
    # accountLoginElement = self.driver(text=accountLoginStr, className="android.view.View")
    accountLoginElement = self.driver(text=accountLoginStr, className="android.view.View",
                                       clickable="true")
    logging.debug("accountLoginElement=%s", accountLoginElement) # accountLoginElement=<uiautomator2._selector.UiObject object at 0x10dd4be80>
    logging.debug("accountLoginElement.exists=%s", accountLoginElement.exists) # accountLoginElement.exists=True
    if accountLoginElement.exists:
        accountLoginElement.click()
        logging.info("Has clicked %s button", accountLoginStr)

    time.sleep(0.1)
else:
    logging.warning("Not found %s button", accountLoginStr)

```

```

# phoneXPath = """//android.widget.EditText[@text="" and @index="5"]"""
phoneXPath = """//android.widget.EditText[@index="5" and @clickable="true" and @focusable="true"]"""
phoneSelector = self.driver.xpath(phoneXPath)
if phoneSelector.exists:
    logging.info("Found 请输入手机号")
    # phoneSelector.set_text(Vivo_Account)
    self.selectorSetText(phoneSelector, self.config["Vivo_Account"])
    logging.info("Has input vivo account phone number")
else:
    logging.warning("Not found 请输入手机号")

passwordStr = "请输入密码"
# passwordXPath = """//android.widget.EditText[@text="%s" and @index="2" and @clickable="true"]""" % passwordStr
passwordXPath = """//android.widget.EditText[@index="2" and @clickable="true" and @focusable="true"]"""
passwordSelector = self.driver.xpath(passwordXPath)
if passwordSelector.exists:
    logging.info("Found %s", passwordStr)
    # pwdClickResp = passwordSelector.click()
    # logging.debug("pwdClickResp=%s", pwdClickResp)
    # doScreenshot(u2Dev)
    self.selectorSetText(passwordSelector, self.config["Vivo_Password"])
    logging.info("Has input vivo password")
else:
    logging.warning("Not found %s", passwordStr)

loginStr = "登录"
# loginXPath = """//android.view.View[@text="登录" and @clickable="true"]"""
loginXPath = """//android.view.View[@text="登录" and @index="12" and @clickable="true"]"""
loginSelector = self.driver.xpath(loginXPath)
if loginSelector.exists:
    loginSelector.click()
    logging.info("Has clicked %s button", loginStr)
else:
    logging.warning("Not found %s", loginStr)

logging.info("Complete auto do vivo account login")

```

- 后记：

- 还会额外弹出验证码手动输入页面
  - 此处无法通过代码获取（另外手机收到的）验证码，所以无法代码自动化，只能手动输入
  - 效果



## 身份验证

验证码 请输入短信验证码 重新获取(51s)

验证码短信已发送至

下一步

短信可能会有延时，请耐心等待，如果长时间未收到短信，  
请60秒后重新获取

若手机号码暂时无法验证，请点击 [手机号已停用？](#)

或联系客服 400-629-9688





## 身份验证

验证码 731162

[获取验证码](#)

验证码短信已发送至

[下一步](#)

短信可能会有延时，请耐心等待，如果长时间未收到短信，  
请60秒后重新获取

若手机号码暂时无法验证，请点击 [手机号已停用？](#)

或联系客服 400-629-9688



- 验证后，会自动返回



crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2021-07-18 09:59:23

## 奇虎360账号自动登录

### 背景

游戏测试期间，遇到很多游戏app，都是来自360应用市场的。

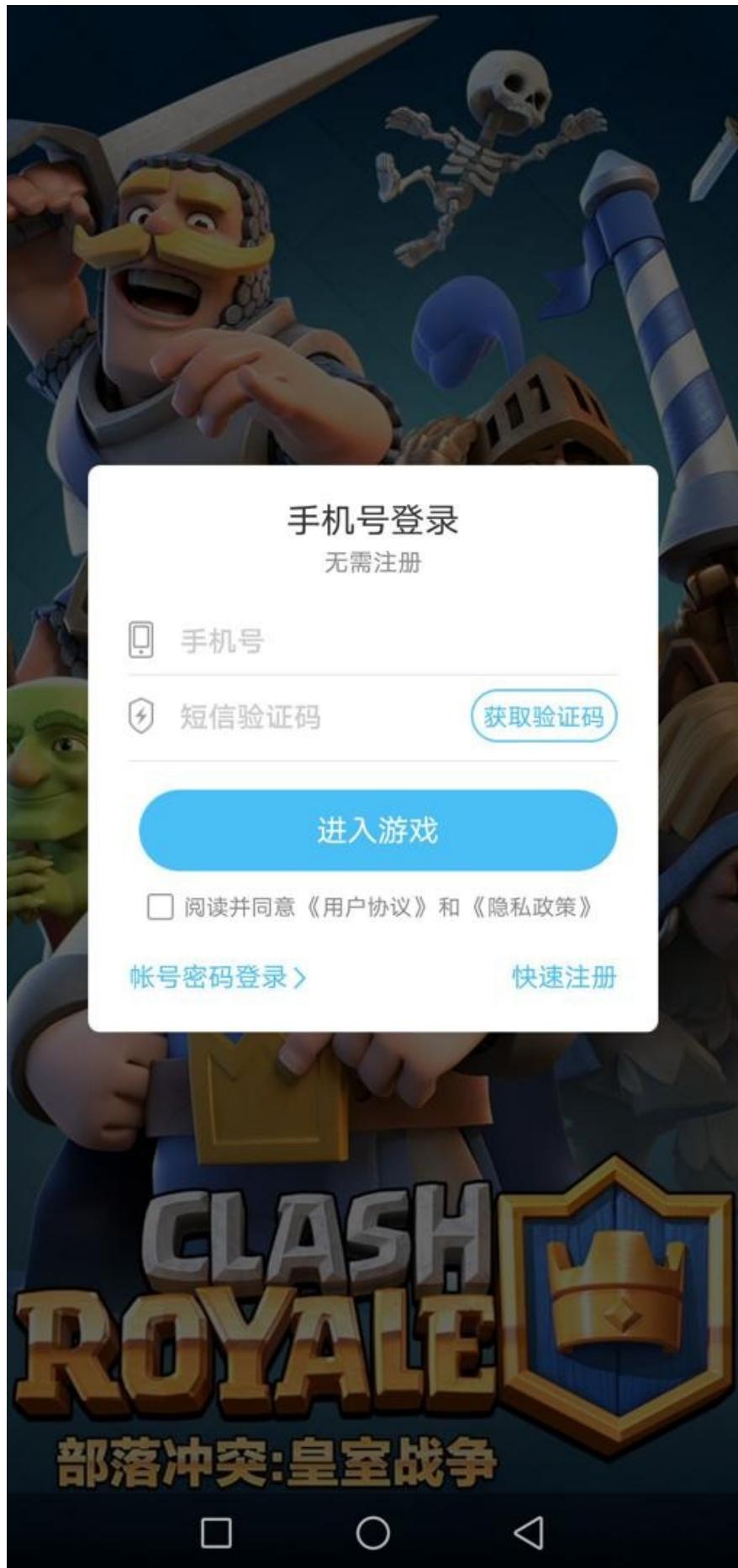
其中在注册和登录期间，会涉及到：

360账号的注册和登录

比如：

游戏：`com.supercell.clashroyale.qihoo_皇室战争`

弹框：



此时：往往需要手动去，切换登录方式，手动输入（已有的）360的账号和密码等操作。比较耗时。

此处用代码实现，自动切换登录方式，输入账号和密码，点击登录。

## 代码

```

Qihoo360_Account: "yourAccount"
Qihoo360_Password: "yourPassword"

# Qihoo360_Login_ReadAndAgree_Xpath: "//android.widget.TextView[@text='阅读并同意' and @resource-id='com.qihoo.gamecenter.pluginapk:id/lpn_checkuseragent_tv1']"
# Qihoo360_Login_ReadAndAgree_Xpath: "//android.widget.TextView[@resource-id='com.qihoo.gamecenter.pluginapk:id/lpn_checkuseragent_userplan']"
Qihoo360_PasswordLogin_Xpath: "//android.widget.TextView[@resource-id='com.qihoo.gamecenter.pluginapk:id/lpn_RegPhone_acount_entry_tv']"

self.driver.watcher.when(Qihoo360_PasswordLogin_Xpath).call(self.autoDo360AccountLogin)

def selectorSetText(self, curXPathSelector, inputText):
    # Special: add click to try workaround for 360 pwd EditText input but input to 360 account EditText
    # curXPathSelector.click()
    # curXPathSelector.clear_text()
    selectorSetTextResp = curXPathSelector.set_text(inputText)
    logging.debug("selectorSetTextResp=%s", selectorSetTextResp) # selectorSetTextResp=None

    # 在set_text后，输入法会变成FastInputIME输入法
    # 用下面代码可以实现：关掉FastInputIME输入法，切换回系统默认输入法
    self.driver.set_fastinput_ime(False)

def autoDo360AccountLogin(self):
    """Auto do 360=qihoo=qihu account login"""
    logging.info("Try auto do qihoo 360 account login")

    # AccountPwdStr = "帐号密码登录"
    AccountPwdStr = "密码登录"
    # reg360AccountPwdXpath = """/android.widget.TextView[@text="%s"]"""\ % AccountPwdStr
    # reg360AccountPwdXpath = """/android.widget.TextView[@resource-id="com.qihoo.gamecenter.pluginapk:id/lpn_RegPhone_exchange_logintype_tv" and @text="%s" and @index="0"]"""\ % AccountPwdStr
    reg360AccountPwdXpath = """/android.widget.TextView[@resource-id="com.qihoo.gamecenter.pluginapk:id/lpn_RegPhone_acount_entry_tv" and @text="%s" and @index="0"]"""\ % AccountPwdStr
    # '//android.widget.TextView[@resource-id="com.qihoo.gamecenter.pluginapk:id/lpn_RegPhone_exchange_logintype_tv" and @text="帐号密码登录" and @index="0"]'
    reg360AccountPwdSelector = self.driver.xpath(reg360AccountPwdXpath)
    if reg360AccountPwdSelector.exists:
        # doScreenshot(u2Dev)
        # logging.info("Found %s", AccountPwdStr)
        reg360AccountPwdSelector.click()
        time.sleep(0.1)

```

```

        logging.info("Has clicked %s button", AccountPwdStr)
    else:
        logging.warning("Not found %s button", AccountPwdStr)

    SwitchLoginTypeStr = "切换登录方式"
    switchLoginTypeXpath = """//android.widget.TextView[@resource-id="com.qihoo.gamecenter.pluginapk:id/lpn_ap_exchange_logintype_tv" and @text="%s" and @index="0"]""" % SwitchLoginTypeStr
    switchLoginTypeSelector = self.driver.xpath(switchLoginTypeXpath)
    isInAccountPwdLoginPage = switchLoginTypeSelector.exists
    if isInAccountPwdLoginPage:
        logging.info("Found %s", SwitchLoginTypeStr)

    if not isInAccountPwdLoginPage:
        ShortSmsVerifyCodeLoginStr = "短信验证码登录"
        # shortSmsVerifyCodeLoginXpath = """//android.widget.TextView[@resource-id="com.qihoo.gamecenter.pluginapk:id/lpn_ap_sms_code_enter_tv" and @text="%s" and @index="0"]""" % ShortSmsVerifyCodeLoginStr
        shortSmsVerifyCodeLoginXpath = """//android.widget.TextView[@resource-id="com.qihoo.gamecenter.pluginapk:id/lpn_ap_sms_code_enter_tv" and @index="0"]"""
        shortSmsVerifyCodeLoginSelector = self.driver.xpath(shortSmsVerifyCodeLoginXpath)
        isInAccountPwdLoginPage = shortSmsVerifyCodeLoginSelector.exists
        if isInAccountPwdLoginPage:
            logging.info("Found %s", ShortSmsVerifyCodeLoginStr)

    if not isInAccountPwdLoginPage:
        AccountLogin360Str = "360帐号登录"
        accountLogin360Xpath = """//android.widget.TextView[@resource-id="com.qihoo.gamecenter.pluginapk:id/lpid_hint_name_tv" and @text="%s" and @index="0"]""" % AccountLogin360Str
        accountLogin360Selector = self.driver.xpath(accountLogin360Xpath)
        isInAccountPwdLoginPage = accountLogin360Selector.exists
        if isInAccountPwdLoginPage:
            logging.info("Found %s", AccountLogin360Str)

    if not isInAccountPwdLoginPage:
        logging.error("Not in 360 account and password login page, can NOT do 360 auto login")
        return

    logging.info("In 360 account and password login page, continue do 360 auto login")

    AccountStr = "360帐号/手机号/邮箱"
    # accountXpath = """//android.widget.EditText[@resource-id="com.qihoo.gamecenter.pluginapk:id/lp_account_input_edit" and @text="%s" and @index="1"]""" % AccountStr
    accountXpath = """//android.widget.EditText[@resource-id="com.qihoo.gamecenter.pluginapk:id/lp_account_input_edit" and @index="1"]"""
    accountSelector = self.driver.xpath(accountXpath)
    if accountSelector.exists:
        logging.info("Found %s", AccountStr)
        # check already input or not
        curAccount = accountSelector.text
        if curAccount and (curAccount == Qihoo360_Account):

```

```

        logging.info("Already inputed 360 account")
    else:
        self.selectorSetText(accountSelector, Qihoo360_Account)
        logging.info("Has input 360 account")
    else:
        logging.warning("Not found %s", AccountStr)

# Specia: 华为Nova 5i此处元素错乱
# 密码输入框 此时的位置 已经变成了 进入游戏 所以 输入密码 会误点击 进入游戏，暂时放弃，待后续研究
# TODO: try xiaomi 10X is or or not
return

PasswordStr = "密码"

# # passwordXPath = """//android.widget.EditText[@resource-id="com.qihoo.gamecenter.pluginapk:id/lp_pwd_input_edit" and @text="%s" and @index="1" and @password="true"]""" % PasswordStr
# # passwordXPath = """//android.widget.EditText[@resource-id="com.qihoo.gamecenter.pluginapk:id/lp_pwd_input_edit" and @index="1" and @password="true"]"""
# # passwordXPath = """//android.widget.EditText[@resource-id="com.qihoo.gamecenter.pluginapk:id/lp_pwd_input_edit" and @index="1"]"""
# passwordXPath = """//android.widget.EditText[@resource-id="com.qihoo.gamecenter.pluginapk:id/lp_pwd_input_edit"]"""
passwordSelector = self.driver.xpath(passwordXPath)
# if passwordSelector.exists:
#     logging.info("Found %s", PasswordStr)
#     curPassword = passwordSelector.text
#     if curPassword and (curPassword == Qihoo360_Password):
#         # logging.info("Already inputed 360 password")
#     else:
#         self.selectorSetText(passwordSelector, Qihoo360_Password)
#         logging.info("Has input 360 password")
# else:
#     logging.warning("Not found %s", PasswordStr)

passwordElement = self.driver(className="android.widget.EditText", resourceId="com.qihoo.gamecenter.pluginapk:id/lp_pwd_input_edit")
logging.info("passwordElement.exists=%s", passwordElement.exists)
if passwordElement.exists:
    passwordElement.set_text(Qihoo360_Password)
    logging.info("Has input 360 password")
else:
    logging.warning("Not found %s", PasswordStr)

AgreeStr = "阅读并同意用户协议"
# agreeeCheckboxElement = u2Dev(className="android.widget.CheckBox", checkable="true", clickable="true")
# if agreeeCheckboxElement.exists:
#     agreeeCheckboxElement.click()
agreeeCheckboxXPath = """//android.widget.CheckBox[@resource-id="com.qihoo.gamecenter.pluginapk:id/l_plugin_checkuseragent_cb" and @index="0" and @clickable="true"]"""
agreeeCheckboxSelector = self.driver.xpath(agreeeCheckboxXPath)

```

```

if agreeCheckboxSelector.exists:
    agreeCheckboxSelector.click()
    time.sleep(0.1)
    logging.info("Has clicked %s checkbox", AgreeStr)
else:
    logging.warning("Not found %s checkbox", AgreeStr)
    return

IntoGameStr = "进入游戏"
IntoGameXpath = """//android.widget.TextView[@resource-id="com.qihoo.gamecenter.plugin.apk:id/lpid_ap_login_btn" and @text="%s" and @index="2" and @clickable="true"]""" % IntoGameStr
loginSelector = self.driver.xpath(IntoGameXpath)
if loginSelector.exists:
    loginSelector.click()
    time.sleep(0.1)
    logging.info("Has clicked %s button", IntoGameStr)
else:
    logging.warning("Not found %s button", IntoGameStr)

# check 请同意用户协议和隐私政策 弹框
PleaseAgreePopupStr = "请同意用户协议和隐私政策"
PleaseAgreePopupXpath = """//android.widget.TextView[@resource-id="com.qihoo.gamecenter.pluginapk:id/txt_msg" and @text="%s" and @index="1"]""" % PleaseAgreePopupStr
PleaseAgreePopupSelector = self.driver.xpath(PleaseAgreePopupXpath)
if PleaseAgreePopupSelector.exists:
    logging.info("Found %s popup", PleaseAgreePopupStr)

PositiveButtonStr = "确定"
positiveButtonXpath = """//android.widget.TextView[@resource-id="com.qihoo.gamecenter.pluginapk:id/btn_pos" and @text="%s" and @index="2" and @clickable="true"]""" % PositiveButtonStr
positiveButtonSelector = self.driver.xpath(positiveButtonXpath)
if positiveButtonSelector.exists:
    positiveButtonSelector.click()
    logging.info("Has clicked %s button", PositiveButtonStr)
    time.sleep(0.1)

# do second time
IntoGameStr = "进入游戏"
IntoGameXpath = """//android.widget.TextView[@resource-id="com.qihoo.gamecenter.pluginapk:id/lpid_ap_login_btn" and @text="%s" and @index="2" and @clickable="true"]""" % IntoGameStr
loginSelector = self.driver.xpath(IntoGameXpath)
if loginSelector.exists:
    loginSelector.click()
    time.sleep(0.1)
    logging.info("Has clicked %s button", IntoGameStr)
else:
    logging.warning("Not found %s button", IntoGameStr)
else:
    logging.info("Not found %s", PleaseAgreePopupStr)

```

```

# wait doing login
LoginingStr = "正在登录..."
loginingXPath = """//android.widget.TextView[@text="%s" and @index="1"]""" % Loginings
try:
    loginSelector = self.driver.xpath(loginingXPath)
    isLogining = loginSelector.exists
    while isLogining:
        loginSelector = self.driver.xpath(loginingXPath)
        isLogining = loginSelector.exists
        logging.info("Is doing login, wait sometime")
        time.sleep(1)

    # locate input cursor to verify code
    VerifyCodeErrorStr = "请输入验证码 (错误码: 5010a)"
    verifyCodeErrorXPath = """//android.widget.TextView[@resource-id="com.qihoo.gamecenter
.pluginapk:id/lpid_hint_error_msg_tv" and @text="%s" and @index="0"]""" % VerifyCodeErrorSt
    r
    verifyCodeErrorSelector = self.driver.xpath(verifyCodeErrorXPath)
    if verifyCodeErrorSelector.exists:
        logging.info("Found %s notice", VerifyCodeErrorStr)

    # locate input cursor
    PleaseInputRightVerifyCodeStr = "请输入右侧的验证码"
    # pleaseInputRightVerifyCodeXPath = """//android.widget.EditText[@resource-id="com.q
    iaho.gamecenter.pluginapk:id/lp_verifycode_input_edit" and @index="1" and @longClickable=
    "true"]"""
    pleaseInputRightVerifyCodeXPath = """//android.widget.EditText[@resource-id="com.q
    iaho.gamecenter.pluginapk:id/lp_verifycode_input_edit" and @index="1"]"""
    pleaseInputRightVerifyCodeSelector = self.driver.xpath(pleaseInputRightVerifyCodeX
    path)
    if pleaseInputRightVerifyCodeSelector.exists:
        pleaseInputRightVerifyCodeSelector.click()
        logging.info("Has clicked %s", PleaseInputRightVerifyCodeStr)
    else:
        logging.warning("Not found %s button", PleaseInputRightVerifyCodeStr)
else:
    logging.warning("Not found %s button", VerifyCodeErrorStr)

```

## 相关截图

### 调试时

weditor截图：

点击 账号和密码登录：

← → C ① localhost:17310

ATX WEditor 0.6.1 Android PWHUT20428002826 Connect Dump Hierarchy 静态 实时

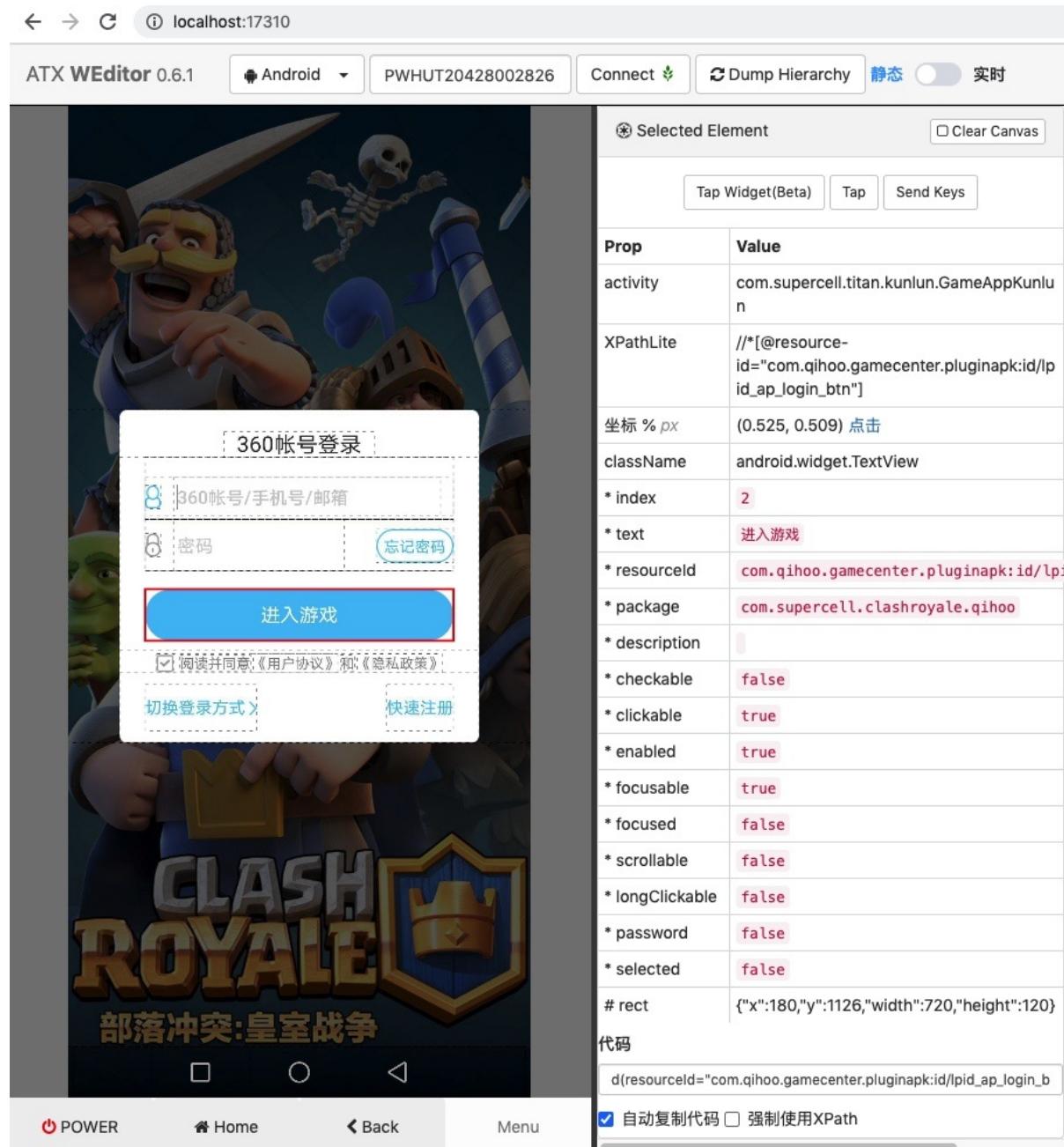
The screenshot shows the Clash Royale login screen. A modal dialog box is displayed, titled "手机号登录" (Phone Number Login) with "无需注册" (No Registration Required). It contains fields for "手机号" (Phone Number) and "短信验证码" (SMS Verification Code), with a "获取验证码" (Get Verification Code) button next to it. Below these fields is a large blue "进入游戏" (Enter Game) button. At the bottom of the dialog, there is a checkbox for accepting terms and conditions and a "账号密码登录" (Account Password Login) link, which is highlighted with a red rectangle. The background features the Clash Royale logo and some characters.

Selected Element  Clear Canvas

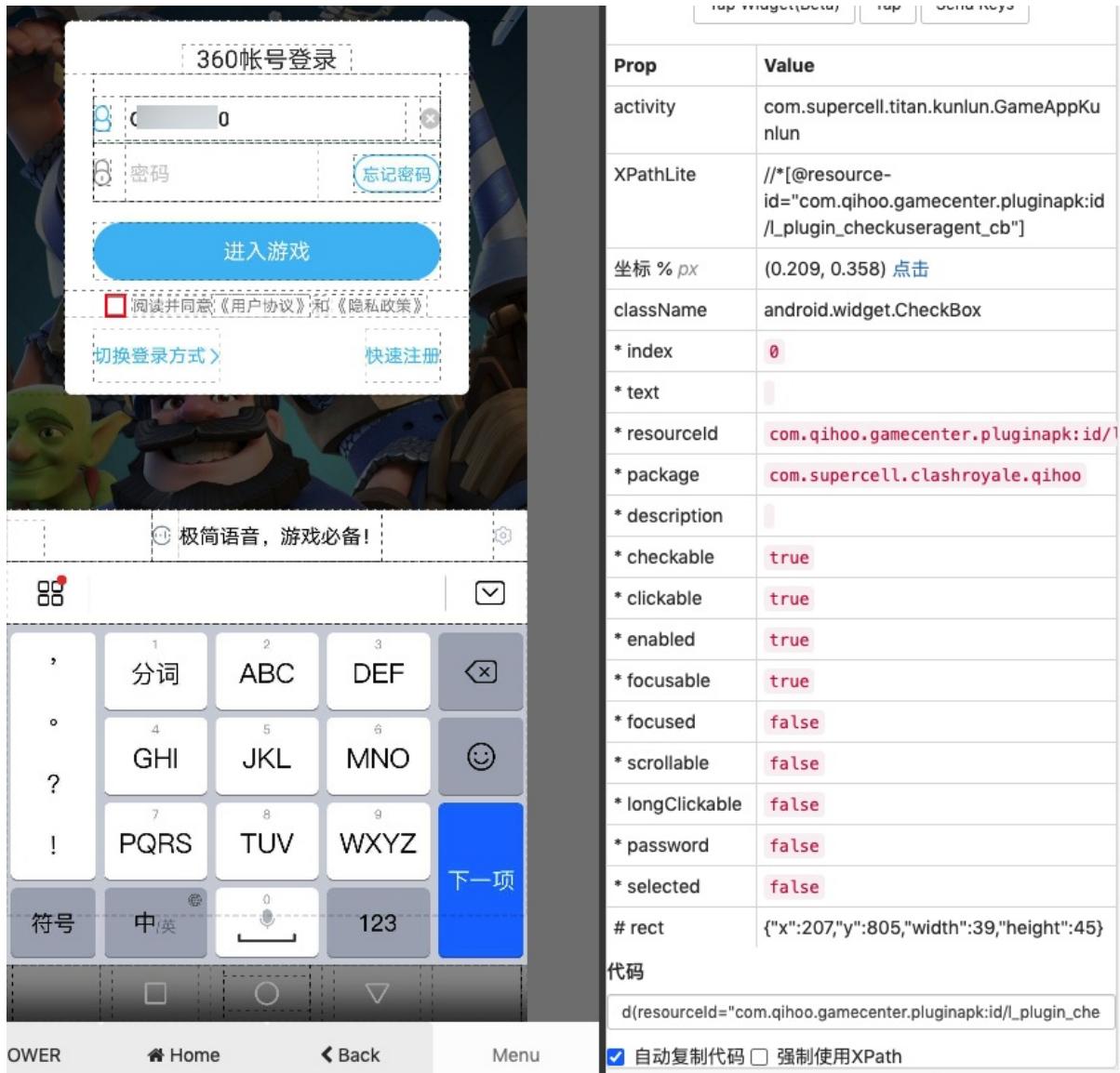
Tap Widget(Beta) Tap Send Keys

| Prop            | Value  |
|-----------------|--|
| activity        | com.supercell.titan.kunlun.GameAppKunlun   |
| XPathLite       | //*[@resource-id="com.qihoo.gamecenter.pluginapk:id/lpn_regphone_exchange_logintype_tv"] |
| 坐标 % px         | (0.305, 0.621) 点击  |
| className       | android.widget.TextView  |
| * index         | 0  |
| * text          | 帐号密码登录   |
| * resourceId    | com.qihoo.gamecenter.pluginapk:id/1  |
| * package       | com.supercell.clashroyale.qihoo  |
| * description   |  |
| * checkable     | false  |
| * clickable     | true   |
| * enabled       | true   |
| * focusable     | true   |
| * focused       | false  |
| * scrollable    | false  |
| * longClickable | false  |
| * password      | false  |

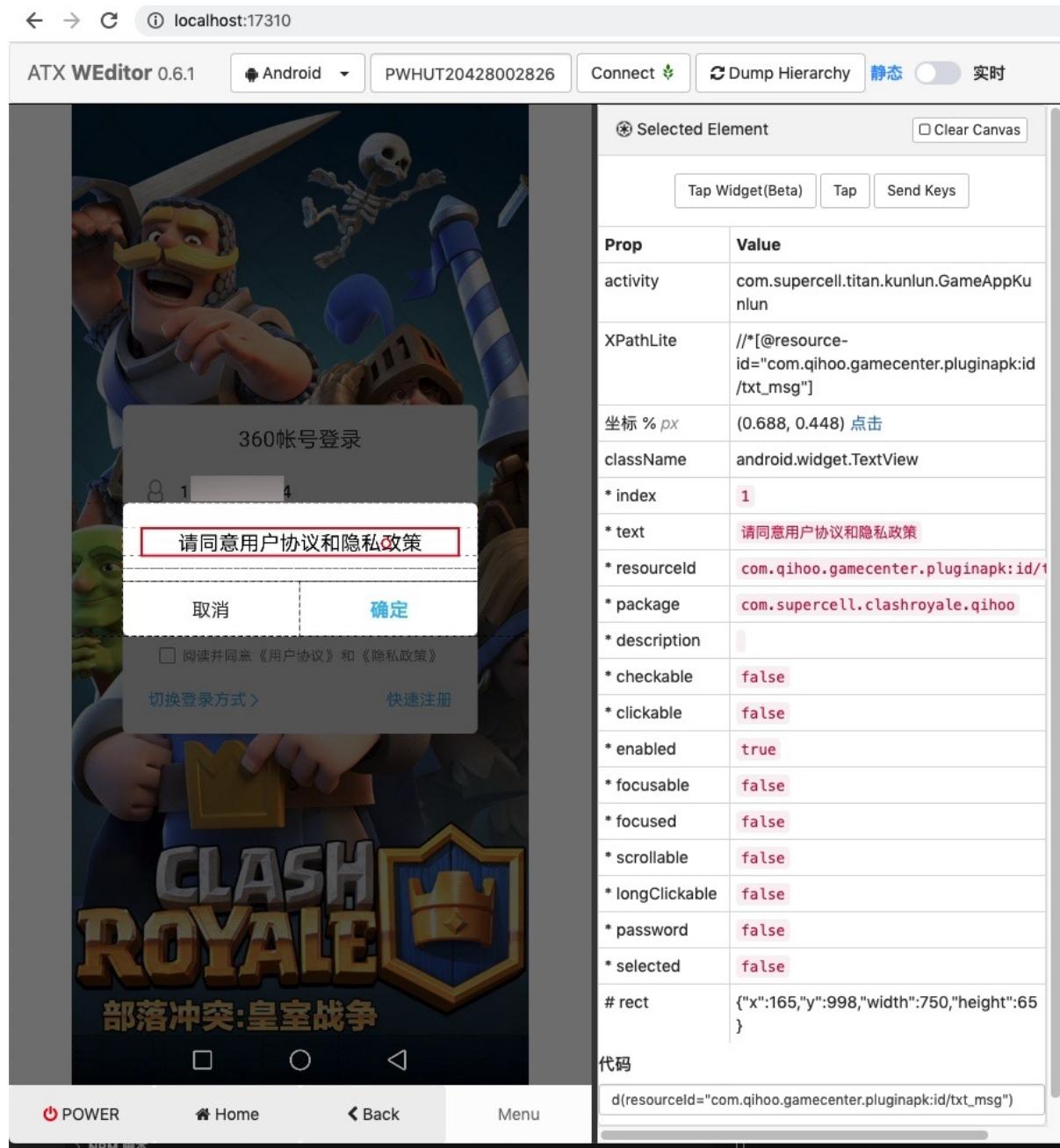
切换到 360账号登录：



已输入360账号：

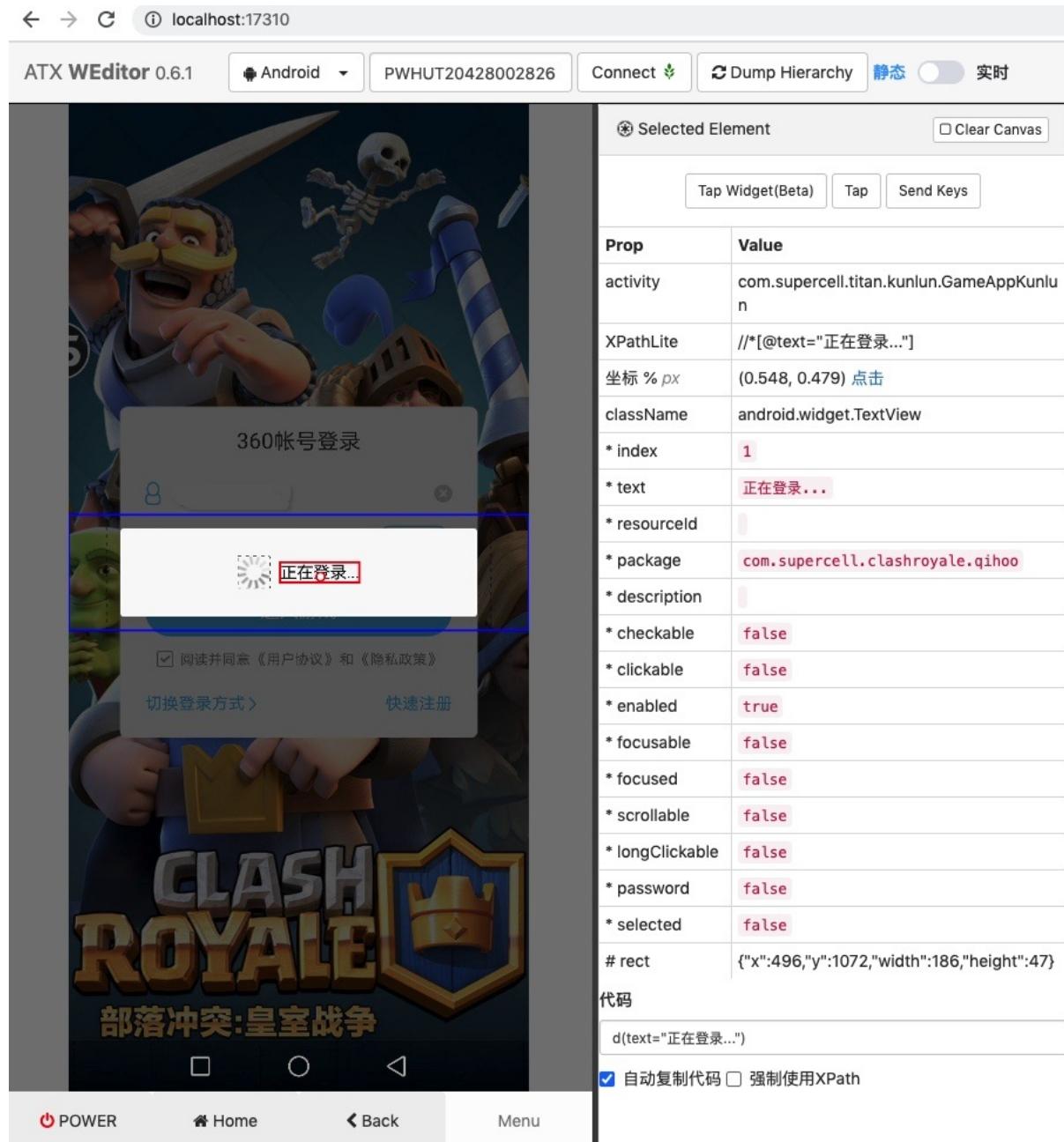


点击登录时，提示：请同意用户协议和隐私政策

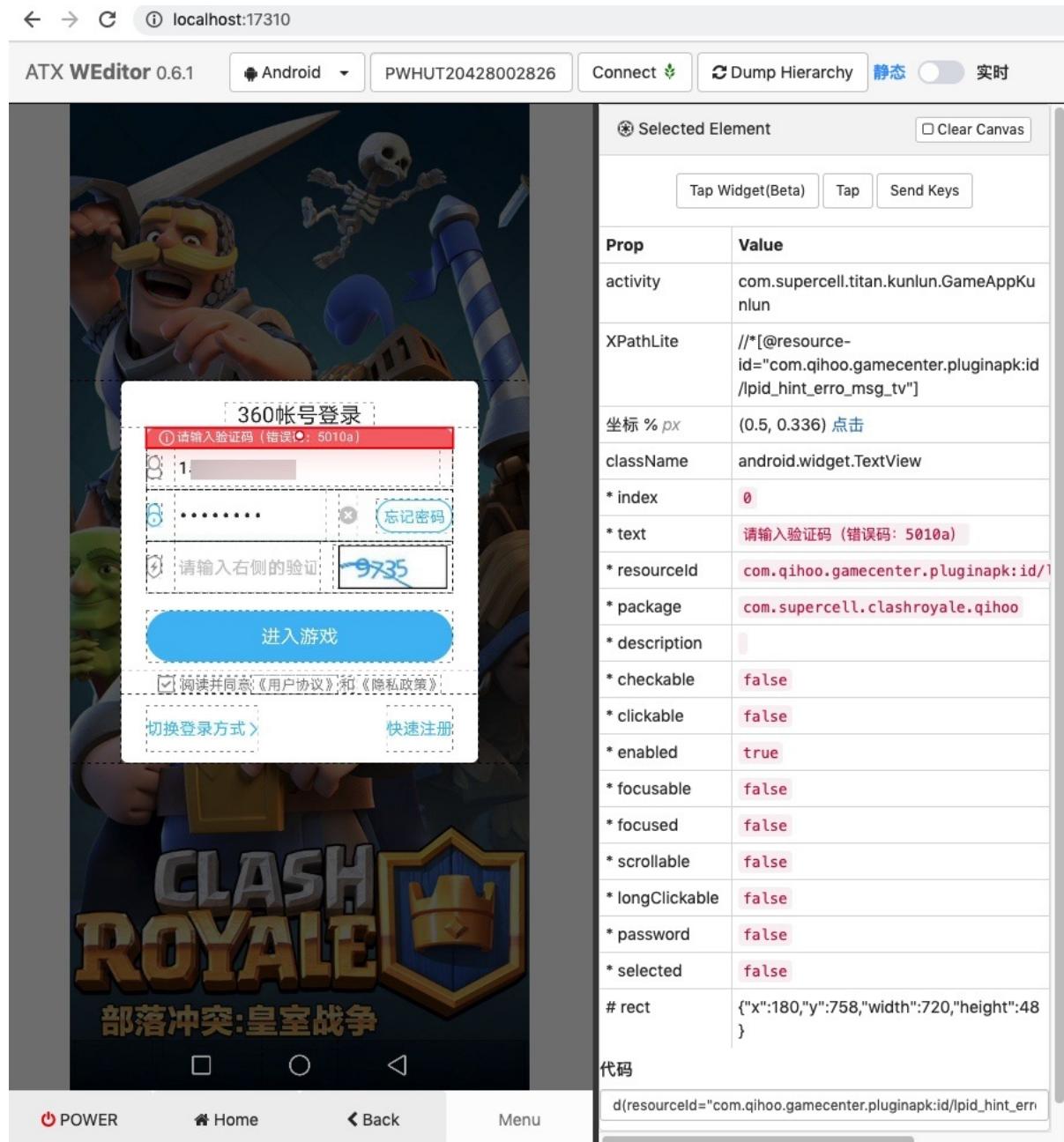


勾线后，点击 进入游戏

往往会提示：正在登录



有时候，经常，会出现错误：请输入验证码



## 实际运行效果

此处实际运行后发现，上述代码，只对于部分360的游戏有效

比如：

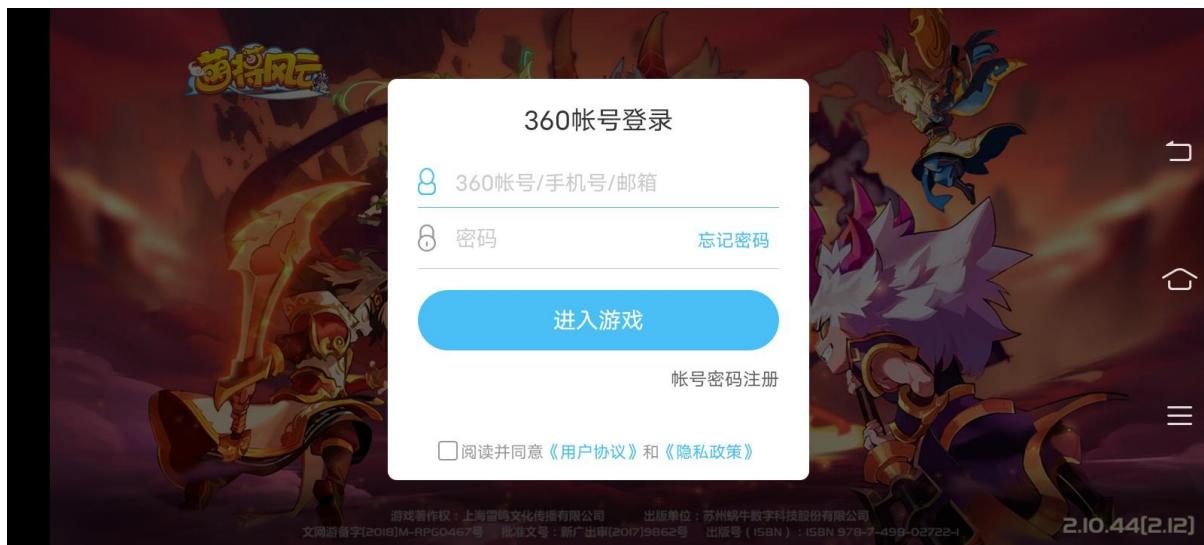
- com.zhw.xzjh.qihoo\_修真江湖

其他还有很多

- 弹框即使一样，但是却也不工作，不起效果的
- 弹框不太一样的

比如：

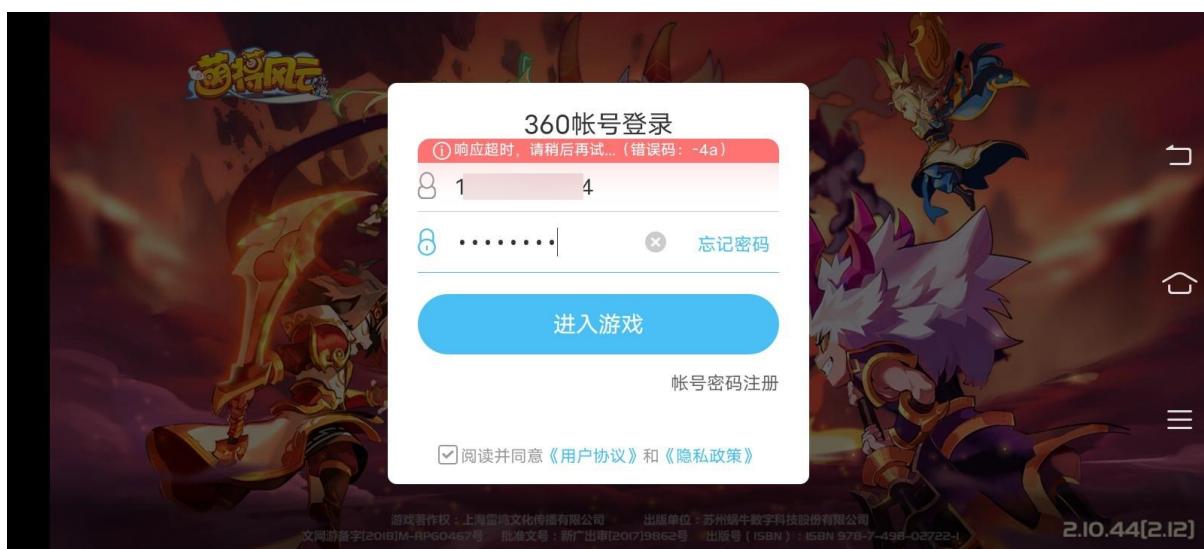
游戏： com.qianhuan.mjfy.qihoo360/萌将风云



-» 都没有 短信密码登录 的按钮

-» 只有 右边才有的 账号密码注册

以及点击登录后，报错也不同：响应超时，请稍后再试 错误码 -4a



红米10X中的游戏 com.mandong.jxqy.qihoo\_剑侠情缘

是旧版本的360账号登录页面：



和游戏 `com.Tq.CQ2ClientAndroid.qihoo/口袋征服` 的：



以及游戏 `com.noumena.android.tinywarcnqh_合金要塞` 的：



新: 2021-07-18 09:59:23

## 附录

下面列出相关参考资料。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09 10:14:28

## 参考资料

- 【记录】 mac中用pipenv安装uiautomator2
- 【未解决】 给安卓手机小米9中欢乐大作战的游戏实现自动挂机
- 【已解决】 红米Note8Pro的uiautomator2初始化出错： OSError Errno Uiautomator started failed
- 【已解决】 uiautomator2获取当前屏幕的宽和高即屏幕大小分辨率信息
- 【已解决】 uiautomator2中如何获取到当前画面的截图文件
- 【部分解决】 python的uiautomator2中set\_text导致输入法变化无法顺利输入文字
- 【已解决】 uiautomator2中点击华为手机中系统自带Swype的输入法中的搜索按钮
- 【已解决】 python的uiautomator2报错： uiautomator2.exceptions.JsonRpcError -32601 Method not found data injectInputEvent
- 【未解决】 uiautomator2中dump\_hierarchy中只能获取到页面的部分的xml源码
- 【已解决】 搞懂uiautomator-server中最终的底层实现dumpWindowHierarchy的处理返回页面数据的逻辑
- 【已解决】 uiautomator2中导出页面源码中NAF是什么意思
- 【未解决】 如何确保uiautomator2的dump\_hierarchy能导出页面中NAF的元素节点
- 【无法解决】 adb发送密码无法解锁安卓手机屏幕
- 【未解决】 自动抓包工具抓包公众号买单吧某个元素通过class+instance定位不到
- 【已解决】 uiautomator2用click点击微信中的通讯录不起作用
- 【已解决】 用weditor实时查看安卓当前页面中的xml源码
- 【已解决】 Mac中安装uiautomator2的UI界面工具： weditor
- 【未解决】 如何修改Android项目android-uiautomator-server的Java代码并重新打包生成2个apk
- 【已解决】 安卓中uiautomator2的set\_text输入导致输入法切换以及恢复输入法
- 【已解决】 红米10X中uiautomator2实现点击Vivo的账号登录切换登录方式
- 【已解决】 uiautomator2给红米10X的vivo账号页自动输入vivo账号和密码
- 【已解决】 自动化游戏测试中合并uiautomator2自动登录vivo账号代码逻辑
- 【已解决】 安卓手机红米10X初始化自动化测试环境
- 【已解决】 红米10X中uiautomator2自动识别Vivo账号登录页面并自动登录
- 【未解决】 自动化测试游戏： 自动实现360账号登录弹框检测和登录流程
- 【已解决】 uiautomator2给小米安全键盘中输入字符串
- 【已解决】 安卓手机小米的红米10X中关闭小米安全键盘输入
- 【已解决】 红米10X中把ATX和com.github.uiautomator:test设置为后台运行不被杀掉
- 【已解决】 设置Vivo安卓手机中运行ATX等应用后台持续运行而不会被进程管理杀掉
- 【已解决】 Mac中用uiautomator2自动操作安卓手机浏览器实现百度搜索
- 【已解决】 Mac中安装和初始化weditor
- 【已解决】 Mac中初始化安卓手机红米Note10X的uiautomator2开发环境
- 【已解决】 安卓手机中查看app应用信息中的包名信息
- 【已解决】 搞清楚安卓手机中多款浏览器应用的包名
- 【已解决】 用uiautomator2获取当前运行的app的包名等信息
- 【已解决】 借助weditor查看安卓当前app包名等信息
- 【已解决】 用uiautomator2查找不到元素如何等待app启动完毕
- 【规避解决】 用uiautomator2操作安卓浏览器新建窗口并打开百度主页

- 【规避解决】 uiautomator2对于安卓浏览器的输入框set\_text输入文字无效
- 【已解决】 用uiautomator2操作安卓浏览器打开百度首页
- 【已解决】 用uiautomator2给安卓浏览器中百度首页中输入文字并触发搜索
- 【已解决】 uiautomator2中dump\_hierarchy获取到的xml源码不是最新的
- 【已解决】 uiautomator2中用xpath如何查找某元素下面的某层级的子元素
- 【基本解决】 uiautomator2中查找和定位安卓浏览器中百度搜索结果中描述文字和来源网站
- 【已解决】 uiautomator2模拟安卓浏览器的百度搜索后去获取和解析搜索结果
- 
- [uiautomator | Android Developers](#)
- [Android 手机自动化测试工具有哪几种？ - 知乎](#)
- [一种 Android 端 Web 多进程情况下支持 Web 自动化测试的方法 - 云+社区 - 腾讯云](#)
- [ATX 文档 - iOS 控件操作 API · TesterHome](#)
- [Manual Init · openatx/uiautomator2 Wiki](#)
- [Android连接常见问题 - Airttest Project Docs](#)
- 

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2021-07-18 09:59:22