目录

安卓自动化测试利器：uiautomator2

- 最新版本：`v2.0`
- 更新时间：`20210330`

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等查找元素、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

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

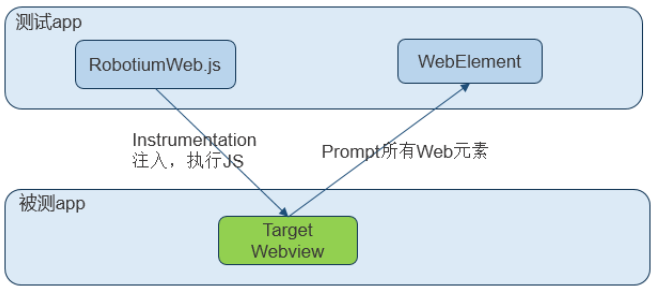
本人 `crifan` 还写了其他 `100+` 本电子书教程，感兴趣可移步至：

crifan/crifan_ebook_readme: Crifan的电子书的使用说明

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

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`

再去安装相关依赖的东西：

```
python3 -m uiautomator2 init
```

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
```

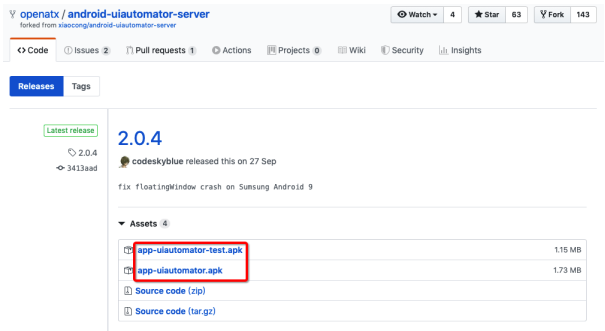
输出举例：

```
➜  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 i
>>> import uiautomator2 as u2
>>> d = u2.connect('8c8a4d4d')
conn=<urllib3.connection.HTTPConnection object at 0x1077f4
```

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httprpc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

- 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.uiautomato
[I 200218 13:56:02 init:300] - app-uiautomator.apk installe
[I 200218 13:56:14 init:300] - app-uiautomator-test.apk ins
[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：

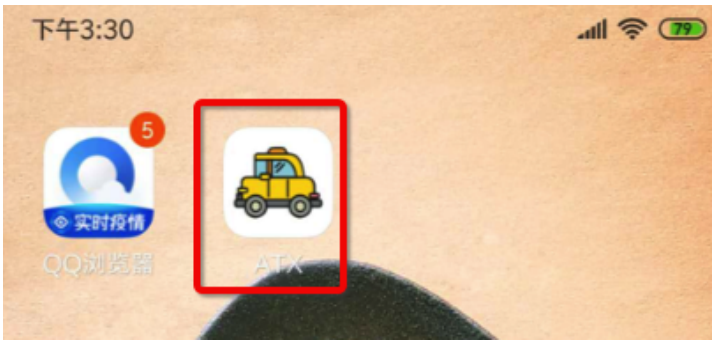
```
[200217 14:45:33][DevicesMethods.py 11 ] start init driver
[I 200217 14:45:37 init:132] uiautomator2 version: 2.5.3
[I 200217 14:45:37 init:317] Install minicap, minitouch
minicap.so |##########| 67.1K/67.1K
[I 200217 14:45:37 init:330] Install com.github.uiautomato
[I 200217 14:45:38 init:300] - app-uiautomator.apk installe
[I 200217 14:45:38 init:300] - app-uiautomator-test.apk ins
[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=hmucaei75ptk7szs)
```

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：

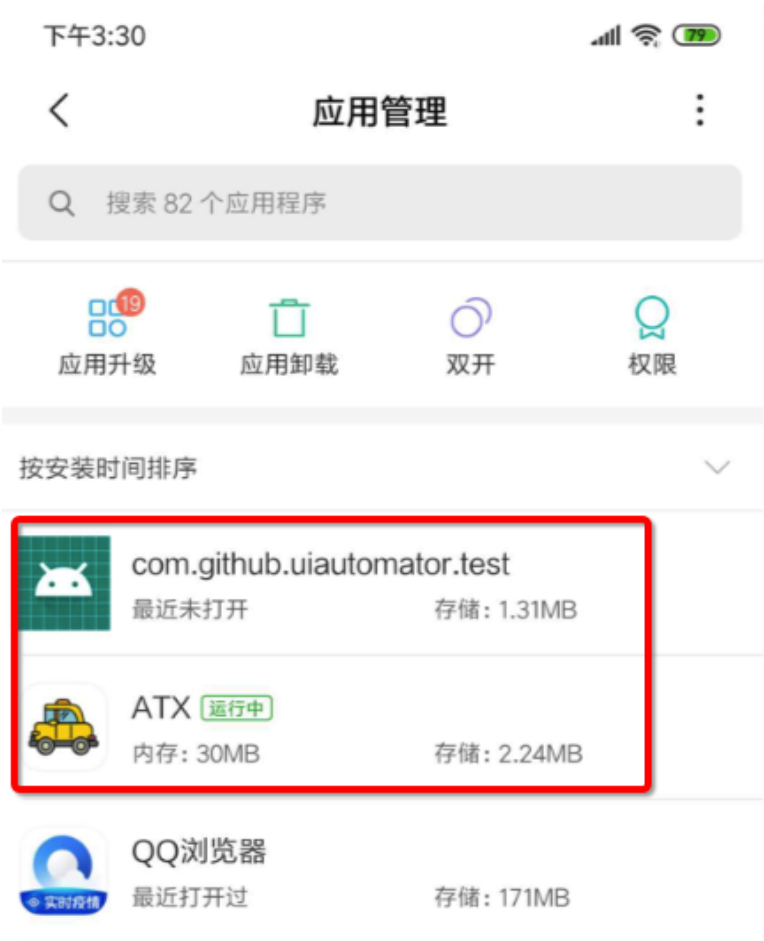
- ATX
  - 桌面图标
    - 

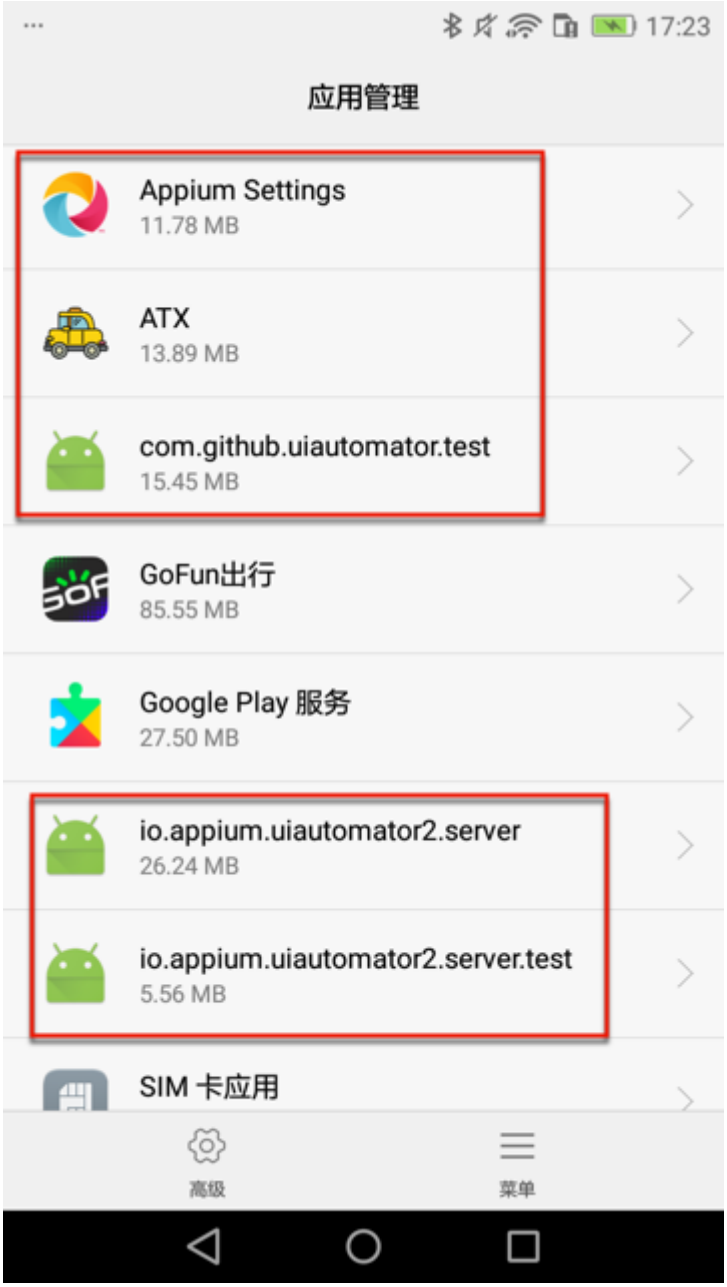
  - 安装期间需要手动点击 继续安装
    - 

- 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`

核心功能

接着介绍uiautomator2的一些常用的核心的功能。

监听

其中一个很常用的，很好用的功能就是：**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 v
  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=ke
    # 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", eachXp

  self.driver.watcher.when(self.config["Vivo_Password_Input
  self.driver.watcher.when(self.config["Vivo_Register_Vivo_
  self.driver.watcher.when(self.config["Permission_Settings
  # self.driver.watcher.when(self.config["Qihoo360_Login_Re
  self.driver.watcher.when(self.config["Qihoo360_PasswordLo

  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）

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。

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常见操作

定位节点和操作节点

```
tbsNodeList = self.driver.xpath("//com.tencent.tbs.core.web
```

获取属性content-desc的值

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

给属性content-desc设置值

```
eachTbsNode.attrib["content-desc"] = "add something to avo
```

删除一个属性

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

文档

关于Xpath的详细用法，见官网中的xpath的文档：

uiautomator2/uiautomator2/ext/xpath at master · openatx/uiautomator2

-》文档已经移至：

uiautomator2/XPATH.md 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
    """
    foundAndClicked = False
    matchDict = {"clickable": "true"}
    clickableParentNode = self.findParentNode(curNodeXpath=
    if clickableParentNode:
        foundNodeAttrib = clickableParentNode.attrib
        clickableParentNode.click()
        foundAndClicked = True
        logging.info("clicked element [%s] found by [xpath=
    else:
        logging.warning("Fail click %s for not found %s(pa

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

                curUpLevel += 1

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

    if not matchNode:
        logging.warning("Not found match parent for xpath=%

    return matchNode
```

isMatchNode：节点是否匹配

```
def isMatchNode(self, curNodeAttrib, toMathInfo):
    """判断当前节点属性是否满足条件"""
    isAllMatch = True
    for eachKey, eachToMatchValue in toMathInfo.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=x
            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
        """
        curTextNodeXpath = "//android.widget.TextView[@text
        self.findAndClickNode(curTextNodeXpath)
```

xpathFindElement：用xpath查找元素

```
def xpathFindElement(self, curClass=None, curId=None, curBo
    """
        find element by xpath

        return value type
            is: u2.xpath.XMLElement
            not: u2.session.UiObject
    """
    foundElement = None
    curXpath = self.generateElementXpath(curClass=curClass,
    try:
        foundElement = self.driver.xpath(curXpath).get()
    except XPathElementNotFoundError as xpathNotFoundErr:
        logging.error("XPathElementNotFoundError: %s from %

    return foundElement
```

调用：

（1）

```
foundElement = self.xpathFindElement(curClass=locatorClass,
```

相关函数：

generateElementXpath：生成元素xpath

```
def generateElementXpath(self, curClass=None, curId=None, c
    """generate element xpath"""
    # nodeXpath = ""
    # if locatorClass:
    #     nodeXpath =  "//%s[@bounds='%s']" % (locatorClass
    # elif locatorId:
    #     nodeXpath =  "//*[@resource-id='%s' and @bounds=
    # 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

    if curBounds:
        if propertyRule:
            propertyRule += " and "

        propertyRule += "@bounds='%s'" % curBounds
        # "@resource-id='com.netease.newsreader.activity:id

    # TODO: add other support: text, desc, instance, ...
    curXpath = "//%s[%s]" % (classRule, propertyRule)
    # "//android.widget.ImageView[@resource-id='com.netease

    return curXpath
```

调用：

```
curClassname = None
curResId = None
curBoundsStr = None

# curAttrib = foundElement.attrib
# AttributeError: 'UiObject' object has no attribute 'attr
if hasattr(foundElement, "attrib"):
    curAttrib = foundElement.attrib
    # {'index': '0', 'text': '', 'resource-id': 'com.neteas
    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, 'to
if not curClassname:
    curClassname = curInfo["className"] # 'android.widget.I

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.n

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=loca
        elif locatorType == "id":
            foundElement = self.driver(resourceId=locat
        elif locatorType == "id+bounds":
            foundElement = self.xpathFindElement(curCla
        elif locatorType == "class+bounds":
            foundElement = self.xpathFindElement(curCla
        elif locatorType == "id+text":
            foundElement = self.driver(resourceId=locat
        elif locatorType == "id+desc":
            foundElement = self.driver(resourceId=locat
        elif locatorType == "class+instance":
            # foundElement = self.driver(className=loca
            instanceInt = int(locatorInstance)
            # foundElement = self.driver(className=loca
            foundElementList = self.driver(className=lo
            if foundElementList:
                curIdx = instanceInt
                shouldMaxNumber = curIdx + 1
                if foundElementList.count >= shouldMaxN
                    foundElement = foundElementList[cur

    if not foundElement:
        if locatorText:
            foundElement = self.driver(text=locatorText
        elif locatorClass:
            foundElement = self.driver(className=locato
        elif locatorDesc:
            foundElement = self.driver(description=loca
        elif locatorId:
            foundElement = self.driver(resourceId=locat
        elif locatorInstance:
            foundElement = self.driver(instance=locator
        elif locatorBounds:
```

```
                foundElement = self.xpathFindElement(curBou

        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, curRig
    # foundElement = self.xpathFindElement(curClass=EditTex
    editTextLocator = {
        "type": "class+bounds",
        "class": EditTextClass, # 'android.widget.EditText
        "bounds": curBounds, # '[939,423][1621,558]'
    }
    foundElement = self.find_element_Android(editTextLocato
    # Note: Not use follow wait_element_setText, for it on
    # isSetOk = self.wait_element_setText(editTextLocator,
    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

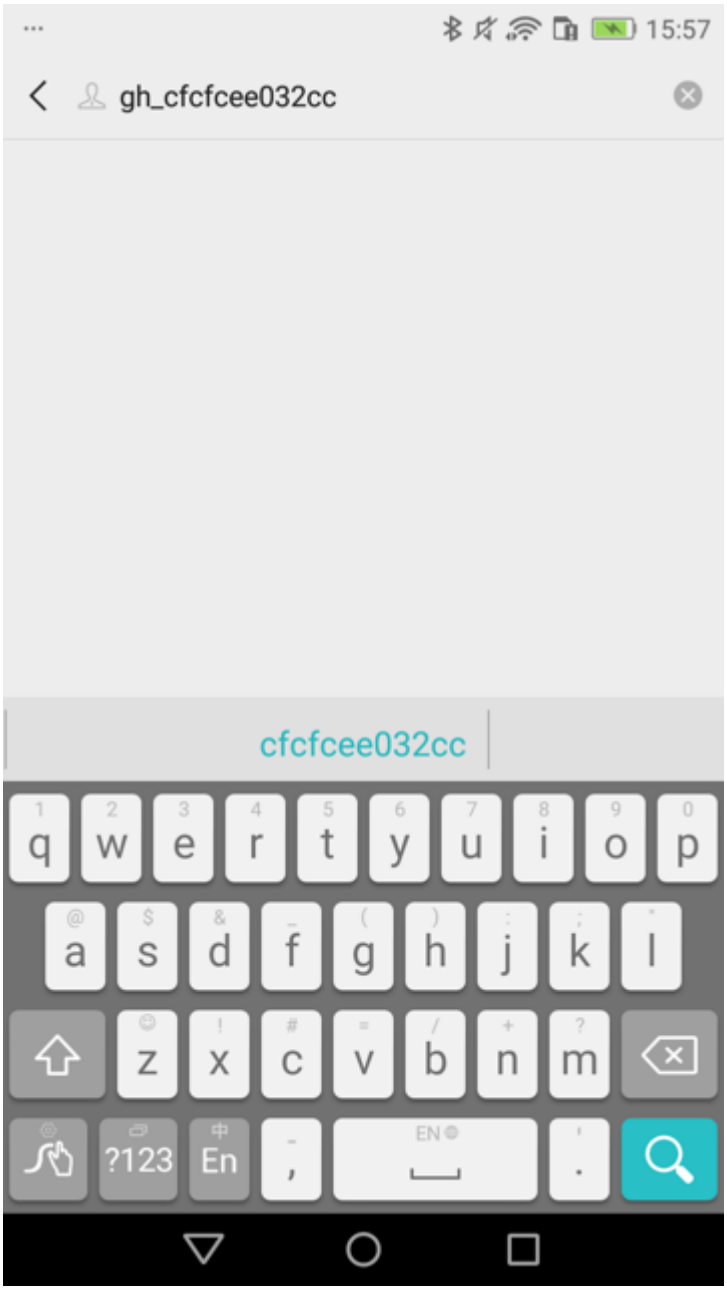
下面详细介绍如何操作。

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=loca
            elif locatorType == "id":
                foundElement = self.driver(resourceId=locat
            elif locatorType == "id+bounds":
                foundElement = self.xpathFindElement(curCla
            elif locatorType == "id+text":
                foundElement = self.driver(resourceId=locat
            elif locatorType == "id+desc":
                foundElement = self.driver(resourceId=locat
            elif locatorType == "class+instance":
                # foundElement = self.driver(className=loca
                instanceInt = int(locatorInstance)
                # foundElement = self.driver(className=loca
                foundElementList = self.driver(className=lo
                if foundElementList:
                    curIdx = instanceInt
                    shouldMaxNumber = curIdx + 1
                    if foundElementList.count >= shouldMaxN
                        foundElement = foundElementList[cur

                if not foundElement:
                    if locatorText:
                        foundElement = self.driver(classNam

                if not foundElement:
                    if locatorBounds:
                        # # method 1: click center point
                        # centerPoint = self.boundsToCenter
                        # self.tap(centerPoint)
```

```
                        # hasClicked = True

                        # method 2: find by xpath with bou
                        foundElement = self.xpathFindElemer
                elif locatorType == "point":
                    centerPoint = self.boundsToCenterPoint(loca
                    self.tap(centerPoint)
                    hasClicked = True
                    foundAndClicked = True
            else:
                if locatorText:
                    foundElement = self.driver(text=locatorText
                elif locatorDesc:
                    foundElement = self.driver(description=loca
                elif locatorId is not None:
                    foundElement = self.driver(resourceId=locat
                elif locatorClass and locatorInstance:
                    foundElement = self.driver(className=locato
                elif locator.get('wexin_text_matched') is not N
                    # TODO 优化微信公众号异常
                    foundElement = self.driver(textContains=loc

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

                if isinstance(foundElement, u2.xpath.XMLElement
                    hasTimeoutPara = False
                    if foundElement:
                        isFound = True
                        isClickableStr = foundElement.attrib["
                        isClickableStr = isClickableStr.lower()
                        if isClickableStr == "true":
                            isClickable = True
                        elif isClickableStr == "false":
                            isClickable = False
                # elif isinstance(foundElement, u2.session.UiOb
                elif isinstance(foundElement, u2.UiObject):
                    hasTimeoutPara = True
                    if foundElement and foundElement.exists:
                        isFound = True
                        if (foundElement.count > 1):
                            foundElement = foundElement[0]

                        isClickable = foundElement.info["clicka

                if isFound:
                    # # for debug: click clickable=false elemer
                    # isClickable = True
```

```
                if isClickable:
                    if hasTimeoutPara:
                        foundElement.click(timeout=wait)
                    else:
                        foundElement.click()
                    foundAndClicked = True
                else:
                    logging.info("Try auto find and click c
                    logging.debug("clickable=false element

                    # curClassName = foundElement.info["cla
                    # curNodeXpath = None
                    # if locatorText:
                    #     # most case: android.widget.TextV
                    #     # sometime: android.view.View
                    #     curNodeXpath = "//%s[@text='%s']"
                    # elif locatorId:
                    #     # {'bounds': '[41,323][1039,727]
                    #     curNodeXpath = "//%s[@resource-ic

                    curClassname = None
                    curResId = None
                    curBoundsStr = None

                    # curAttrib = foundElement.attrib
                    # AttributeError: 'UiObject' object has
                    if hasattr(foundElement, "attrib"):
                        curAttrib = foundElement.attrib
                        # {'index': '0', 'text': '', 'resou
                        curResId = curAttrib["resource-id"]
                        curBoundsStr = curAttrib["bounds"]
                    else:
                        # # for debug
                        # self.debugPrintElement(foundEleme
                        logging.debug("")

                    curInfo = foundElement.info
                    # {'bounds': {'bottom': 2134, 'left': 7
                    if not curClassname:
                        curClassname = curInfo["className"]

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

```
                    if not curResId:
                        if "resourceName" in curInfo:
                            curResId = curInfo["resourceNa

                    curNodeXpath = self.generateElementXpa
                        curClass=curClassname,
                        curId=curResId,
                        curBounds=curBoundsStr,
                    )

                    if curNodeXpath:
                        # # for debug
                        # if curResId:
                        #     if re.search("(id/a_9)|(id/b
                        #         self.debugPrintElement(f
                        # else:
                        #     self.debugPrintElement(found

                        foundAndClicked = self.findAndClic
                    else:
                        # TODO: add other type later
                        logging.warning("Not click for cli
                            foundElement.info, locator)
            else:
                logging.warning("Not click for not found e

    time.sleep(wait)
    return foundAndClicked
```

调用：

```
if self.isAndroid:
    foundAndClicked = self.wait_element_click_Android(loca
```

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 Edit
    # curXpathSelector.click()
    # curXpathSelector.clear_text()
    selectorSetTextResp = curXpathSelector.set_text(inputTe
    logging.debug("selectorSetTextResp=%s", selectorSetText
    # 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
    self.driver.set_fastinput_ime(False)
```

调用举例：

```
Qihoo360_Account = "yourAccount"
accountXpath = """//android.widget.EditText[@resource-id="c
accountSelector = self.driver.xpath(accountXpath)
self.selectorSetText(accountSelector, Qihoo360_Account)
```

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
SwipDuration_Android = 0.3

SwipeDirectionBounds = [338, 333, 38, 333]
curSession.swipe(SwipeDirectionBounds[0], SwipeDirectionBou

self.driver.swipe(SwipeDirectionBounds[0], SwipeDirectionBo
```

（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：

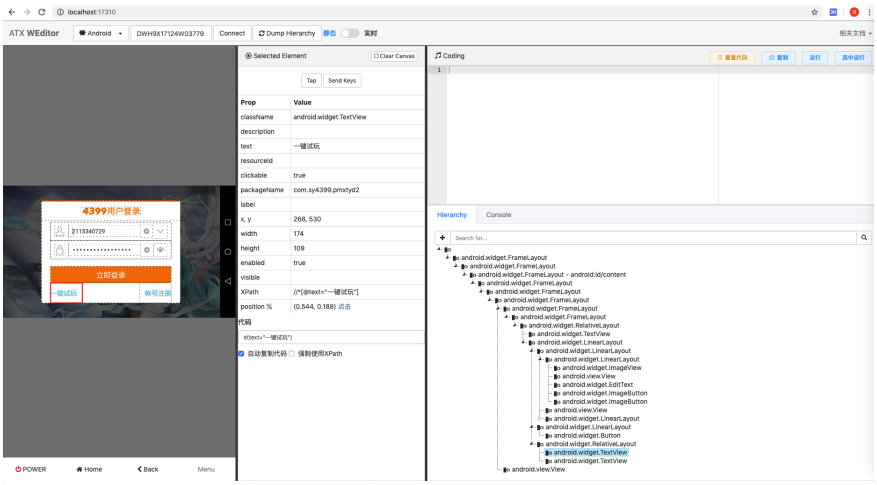
```
def getCurPageSource(self):
    # curPageSrcXml = self.driver.dump_hierarchy()
    curPageSrcXml = self.driver.dump_hierarchy(compressed=F

    # output, exitCode = self.driver.shell(["adb", "shell",
    # output, exitCode = self.driver.shell(["uiautomator",
    # output, exitCode = self.driver.shell("uiautomator dum
    # output, exitCode = self.driver.shell(["shell", "uiaut
    # 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,
```

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.v
...
                                                        <node N
                                                    </node>
                                                </node>
                                                <node index="1"
                                                    <node index
                                                </node>
...
                                                </node>
                                            </node>
                                        </node>
                                    </node>
                                <node index="1" text="" resourc
                            </node>
                        </node>
                    </node>
                </node>
            </node>
    </node>
</hierarchy>
```

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 alread installed".format(a
  else:
    logging.info("start to install app in {}".format(os.pat

    # show current adb command version, makesure is latest
    # os.system("adb --version")
    # Android Debug Bridge version 1.0.41
    # Version 30.0.5-6877874
    # Installed as /Users/limao/dev/tools/android/adb/platf

    # isUseShortCmd = True # use short command to try to av
    isUseShortCmd = False # use default long abd install co

    if isUseShortCmd:
      # sometime will stuck, reason: maybe command length t
      # so copy to temp folder -> to reduce command string
      tmpFolder = tempfile.TemporaryDirectory()
      logging.info("tmpFolder=%s", tmpFolder)
      # tmpFolder=<TemporaryDirectory '/var/folders/gt/5868
      tmpFolderName = tmpFolder.name
      logging.info("tmpFolderName=%s", tmpFolderName)
      # tmpFolderName=/var/folders/gt/5868sbcd1jq4rxvryqhy2
      apkFileName = os.path.basename(apkFilePath)
      logging.info("apkFileName=%s", apkFileName)
      # apkFileName=20201202_fengyun_0192LeiMoChuanShuo_gar
      tmpApkFile = os.path.join(tmpFolderName, apkFileName)
      logging.info("tmpApkFile=%s", tmpApkFile)
      # tmpApkFile=/var/folders/gt/5868sbcd1jq4rxvryqhy2_1s
      logging.info("Copy %s to tmp file %s", apkFilePath, t
      # Copy /Users/limao/dev/xxx/crawler/appAutoCrawler/Ap
      copyfile(apkFilePath, tmpApkFile)
      # copy2(apkFilePath, tmpFolderName)

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

    # show file size
    fileSizeInt = os.path.getsize(curApkFile) # 259106541
    fileSizeStr = CommonUtils.formatSize(fileSizeInt) # '24
```

```
    logging.info("file size: %s", fileSizeStr) # file size:

    # adbInstallCmd = "adb -s {0} install -r {1}".format(se
    # adbInstallCmd = "adb -s {0} install -r {1}".format(se
    # installPara = " "
    # installPara = "-r"
    installPara = "-r -f"
    adbInstallCmd = "adb -s {0} install {1} {2}".format(se
    # adbInstallCmd = "adb shell pm install -s {0} {1}".for
    # adbInstallCmd = "adb shell pm install -s {0} -r {1}".
    # adbInstallCmd = "adb shell pm install -s {0} -f {1}".
    # adb -s hmucaei75ptk7szs install -r /var/folders/gt/58
    # length=243 command:
    # adb -s hmucaei75ptk7szs install -r /Users/limao/dev/x
    # 'adb -s hmucaei75ptk7szs install -r -f /Users/limao/d

    logging.info("Run length=%d, command: %s", len(adbInsta
    # Run length=153, command: adb -s hmucaei75ptk7szs inst
    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, sto
```

卸载app

```
def uninstallApp_Android(self, item):
  # 卸載安裝包
  appPackage = item[1]
  adbUninstallCmd = "adb -s %s uninstall %s" % (self.device
  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 iter
  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()
```

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)
```

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。

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：

```
pip3 install -U weditor
```

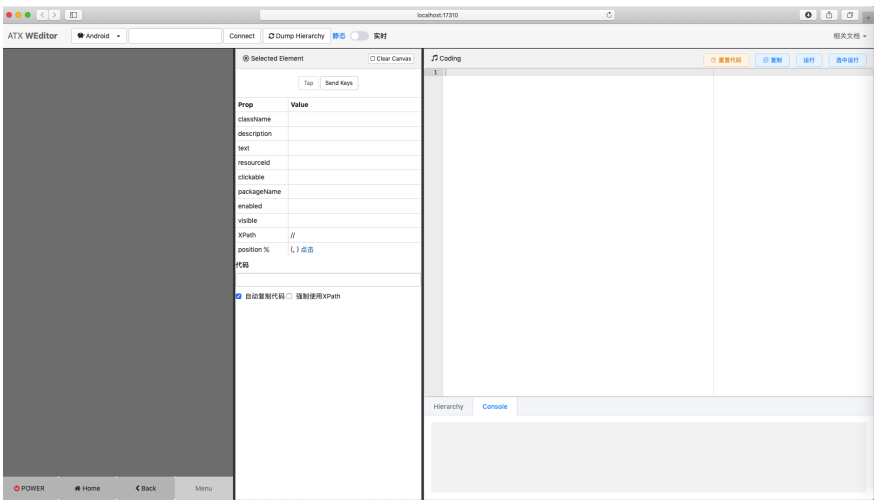
启动：

```
python -m weditor
```

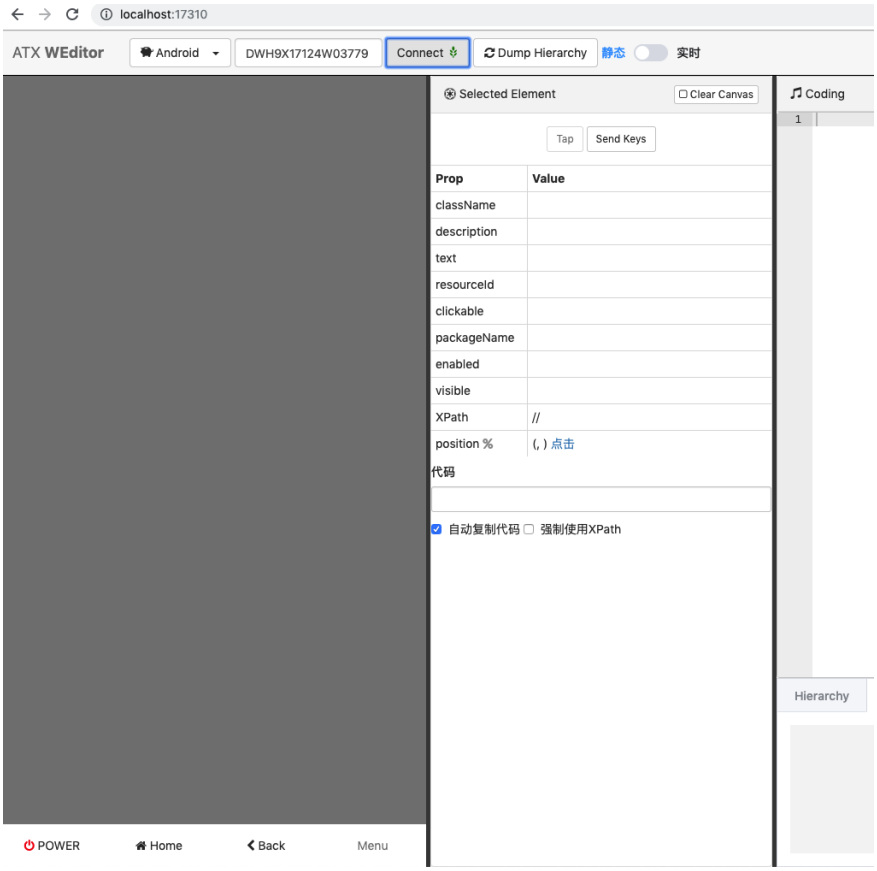
会自动调用浏览器并打开网址：

http://localhost:17310

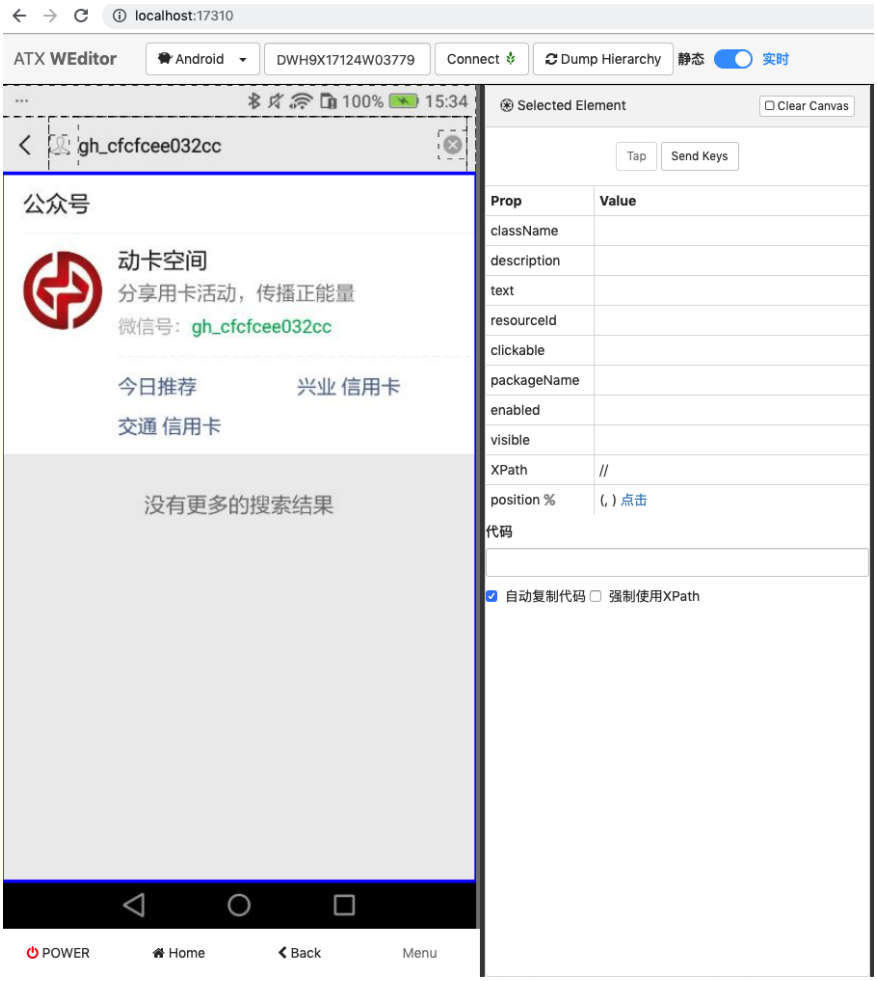
效果：



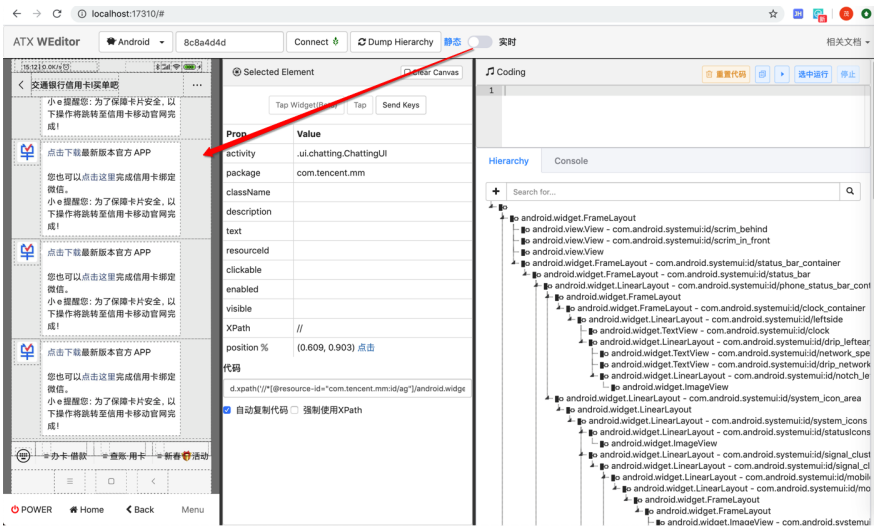
输入 `安卓设备的id` 后，点击 `Connect` 连接设备：



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：

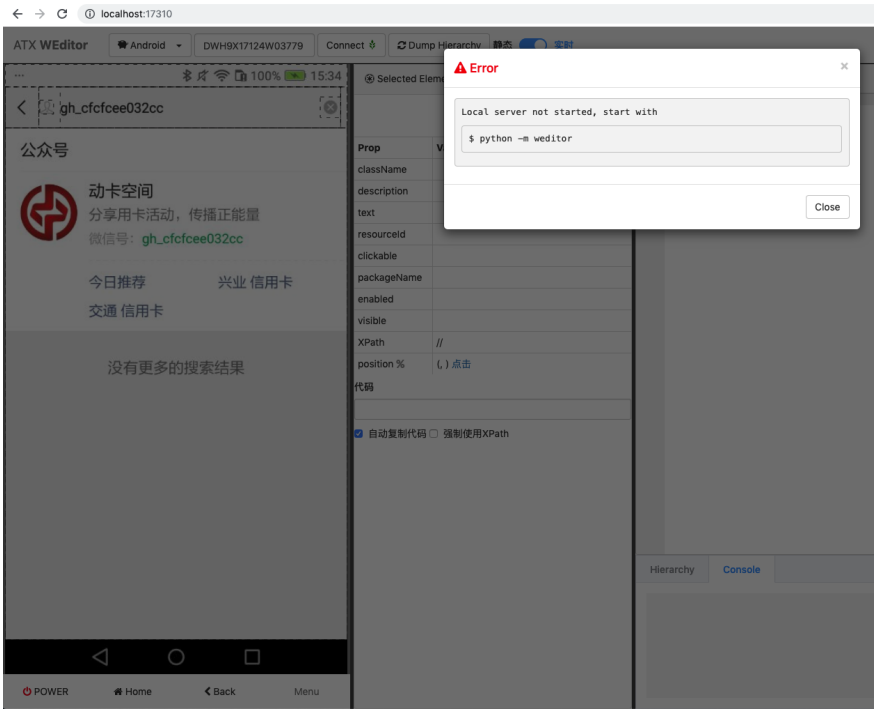


和：



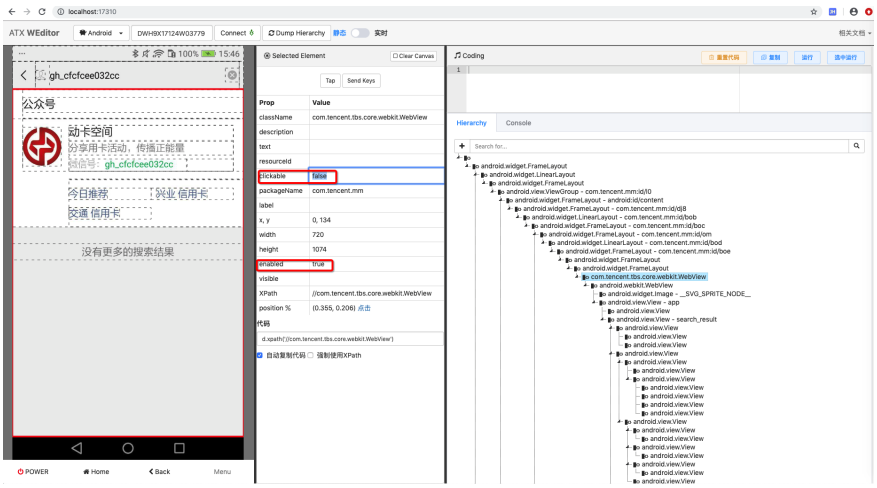
报错可忽略

注意，切换期间偶然会报错：



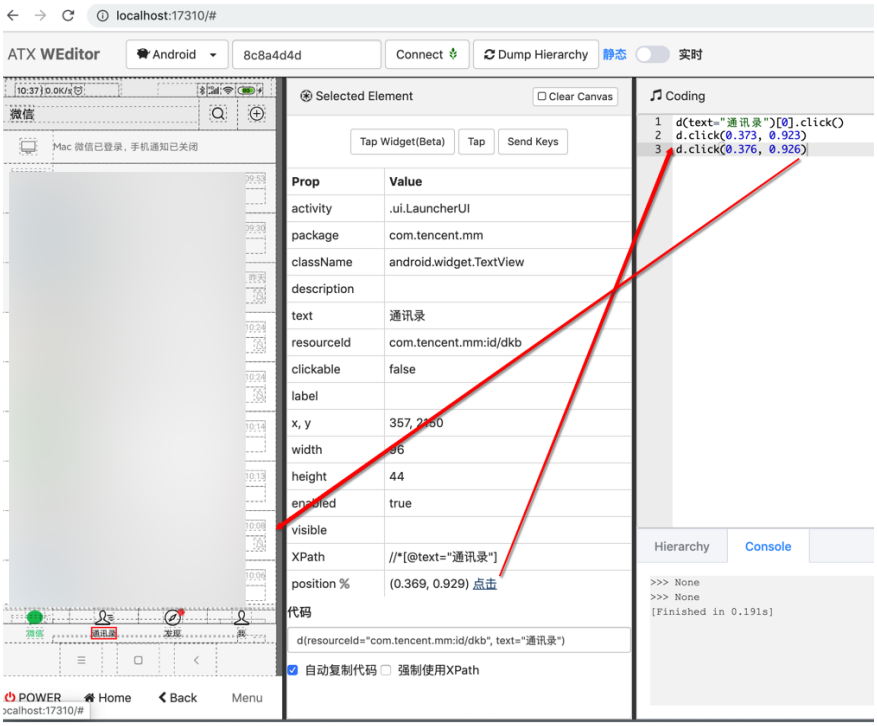
不用理会，关闭弹框，多试几次即可。

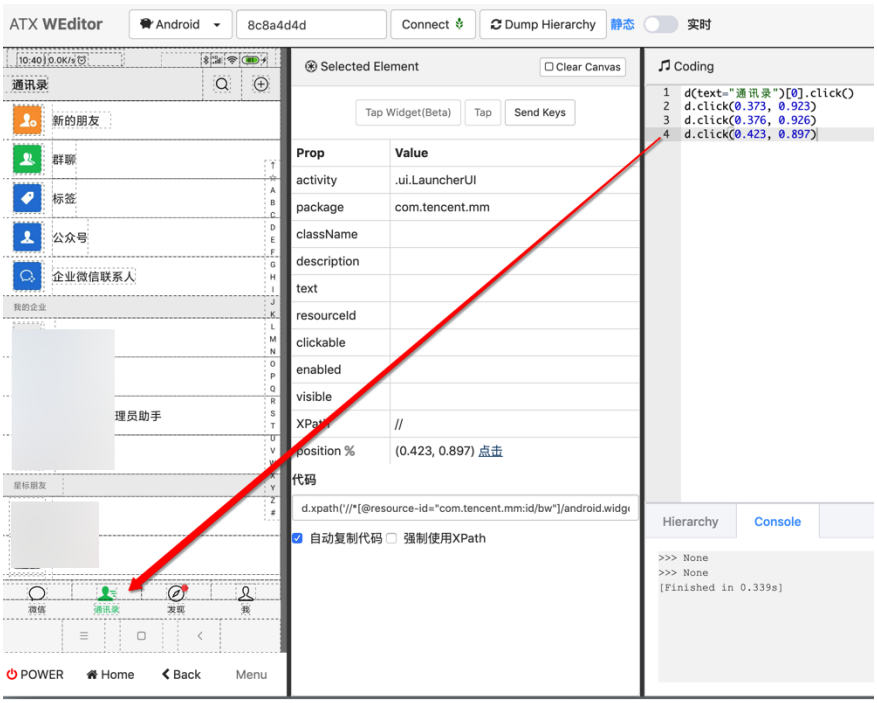
点击 `Dump Hierarchy` 后，能查看到页面的结构：



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：

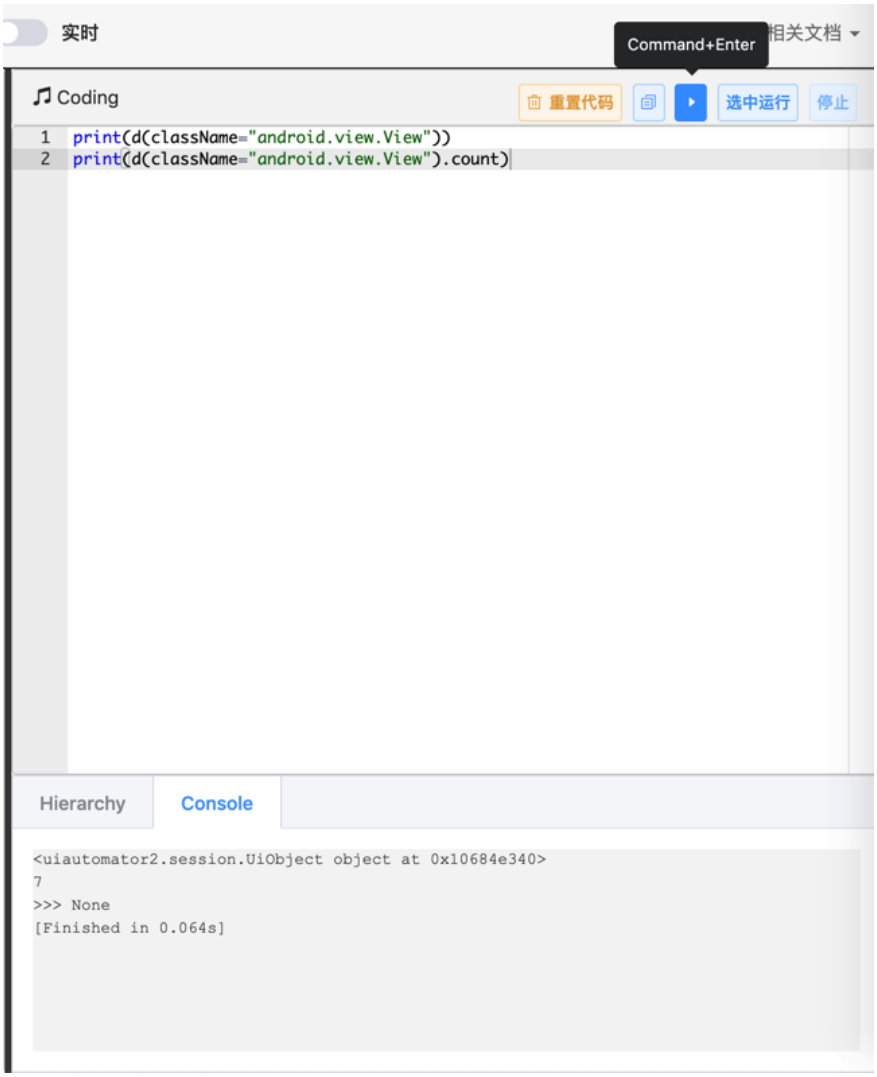
```
d(className="android.view.View")
d(className="android.view.View").count
```

选中第一行后，点击 选中运行：

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：



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
```

想要去WEditor中

搜id值，即搜 `com.tencent.mm:id/pq` ，结果找不到

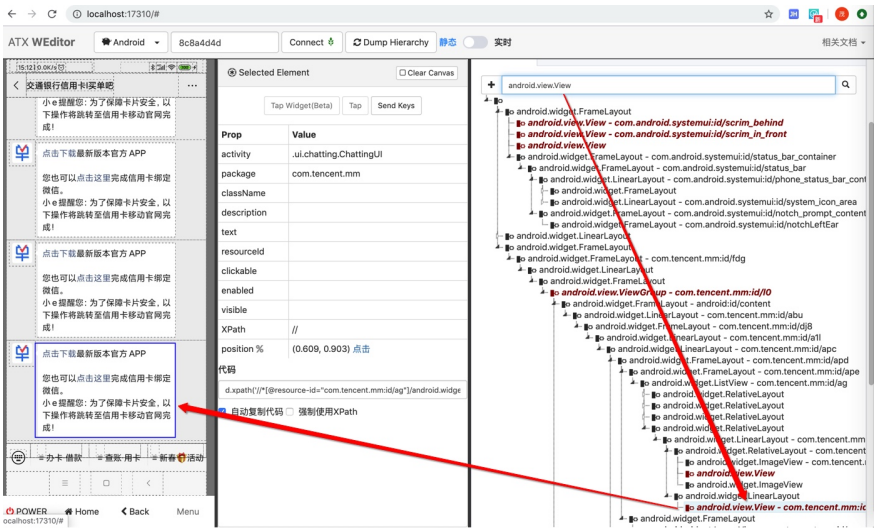
搜pq，也搜不到

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

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
```

或：

```
adb -s DWH9X17124W03779 shell input touchscreen swipe 300 3
```

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

其发布出的

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/bui
    - mv app/build/outputs/apk/androidTest/debug/app-debug-
```

uiautomator

android官网的工具：uiautomator

主页：uiautomator | Android Developers

说了具体用法：

```
adb shell uiautomator dump
```

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。

输入文字

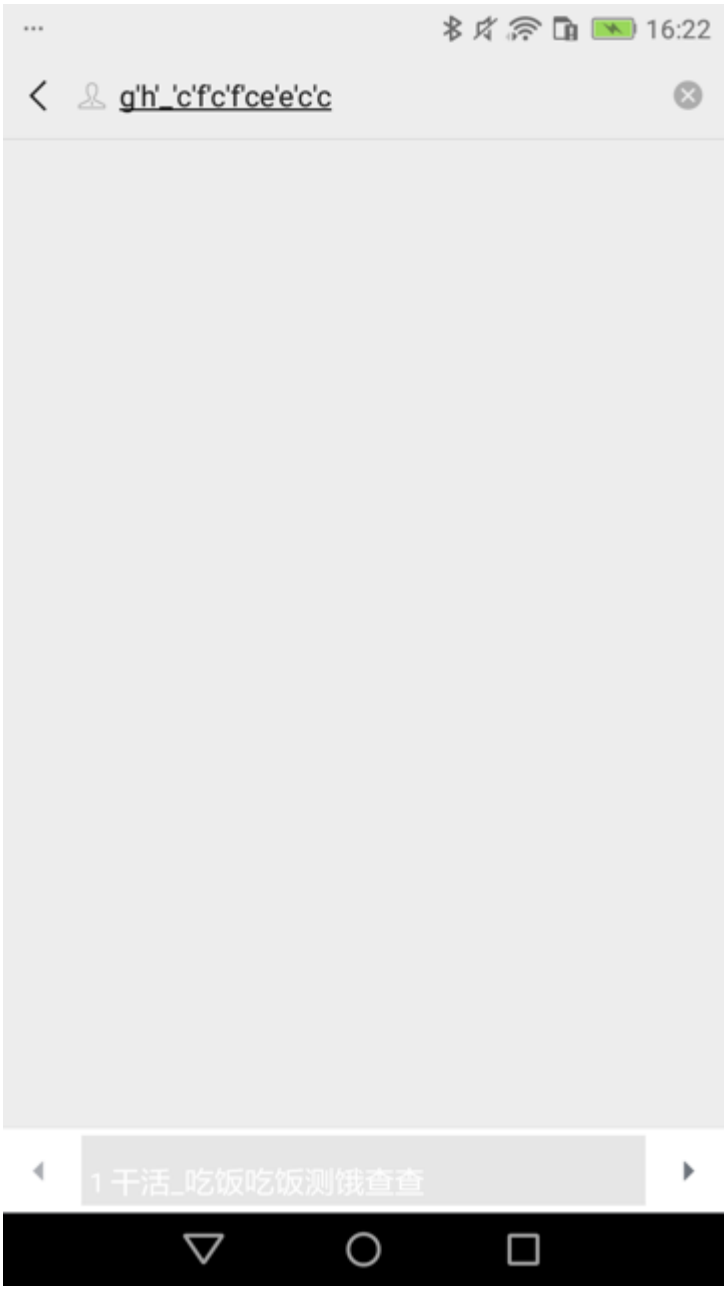
输入文字的两种方式

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=Wa
```

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=Wa
```

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.selecto
        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_fastinput_ime(False) # 关掉FastInputIME输入法
```

其中，是否加上 打开FastInputIME

```
self.driver.set_fastinput_ime(True) # # 切换成FastInputIME输
self.driver.send_keys(text)
self.driver.set_fastinput_ime(False) # 关掉FastInputIME输入法
```

经测试，感觉没区别。

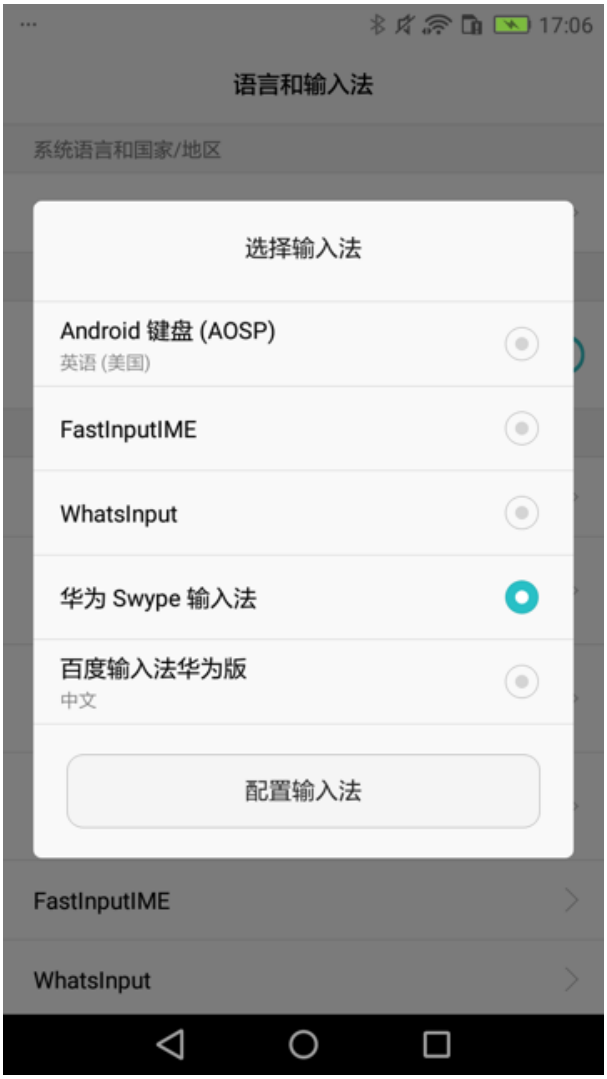
结果都是：

- 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

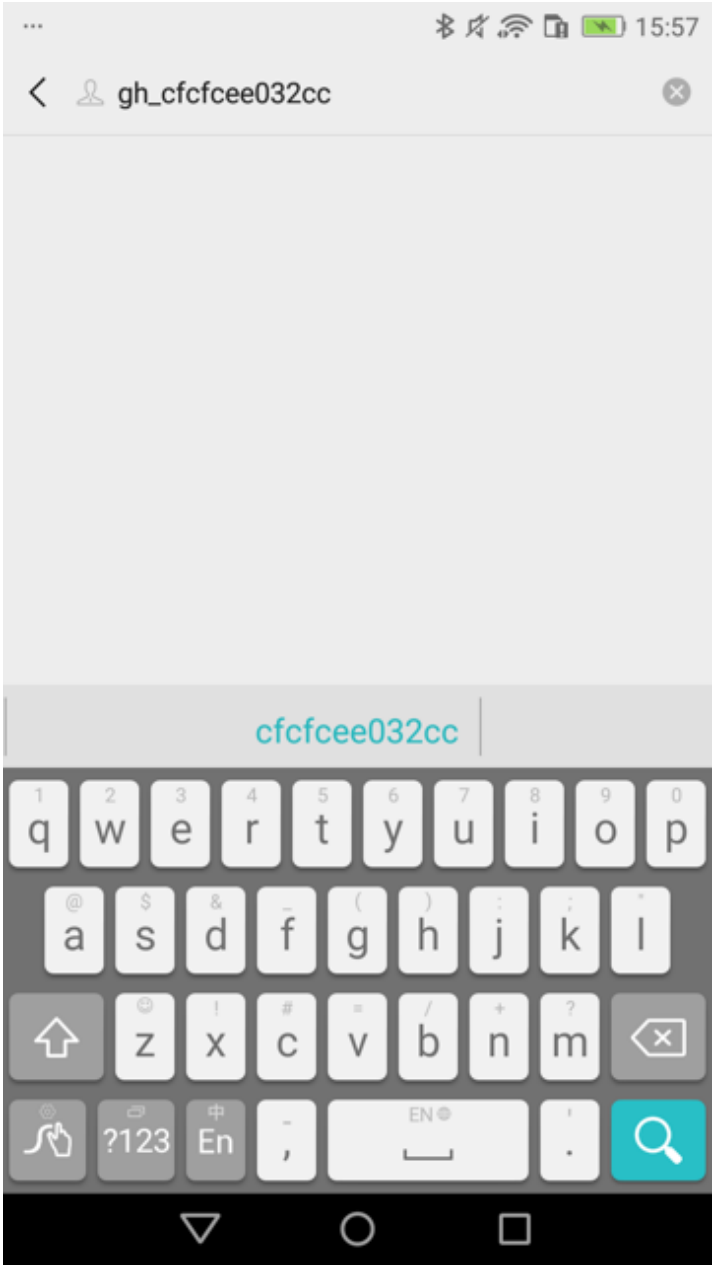
    ```
    * ![android_input_method_baidu](../assets/img/and
    ```

    - 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


17:06
语言和输入法
系统语言和国家/地区
选择输入法
Android 键盘 (AOSP)
英语 (美国)
FastInputIME
WhatsInput
华为 Swype 输入法
百度输入法华为版
中文
配置输入法
FastInputIME
WhatsInput

- 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(inputTe
    logging.info("selectorSetTextResp=%s", selectorSetTextR
    # 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
    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
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(inputTe
    logging.info("selectorSetTextResp=%s", selectorSetTextR
    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
    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

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 la
        # here reset uiautomator2 is workaround for later c
        # Note2: 小米9 Android 10 / 红米 Note8 Pro Android 9
        isNeedRestartU2 = False

        curAndroidVersionFloat = self.getAndroidVersion()
        ANDROID_VERSION_NEED_RESTART_U2 = 7.0
        if curAndroidVersionFloat <= ANDROID_VERSION_NEED_R
            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

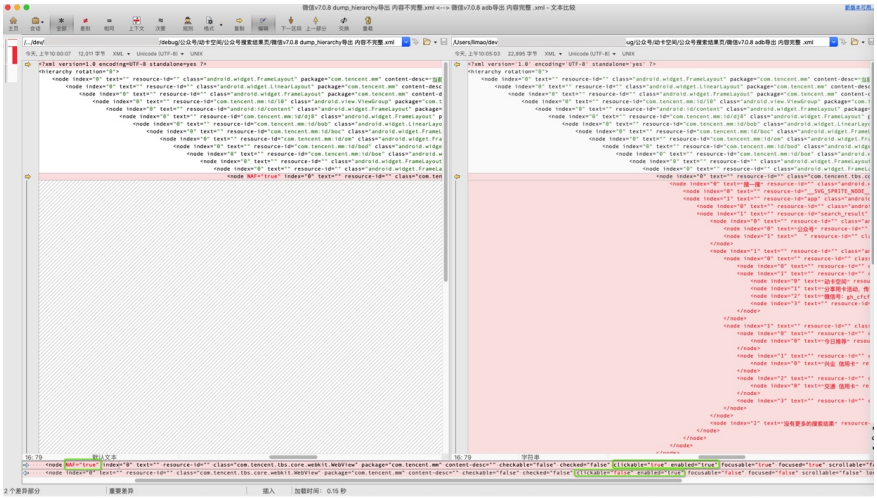
- Mac中终端运行adb命令：`adb shell uiautomator dump`
  - 能导出完整页面的源码

核心代码差异是：

```
<node NAF="true" index="0" text="" resource-id="" class="co
```

和

```
<node index="0" text="" resource-id="" class="com.tencent.t

<node index="0" text="搜一搜" resource-id="" class="android.
...
```



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：

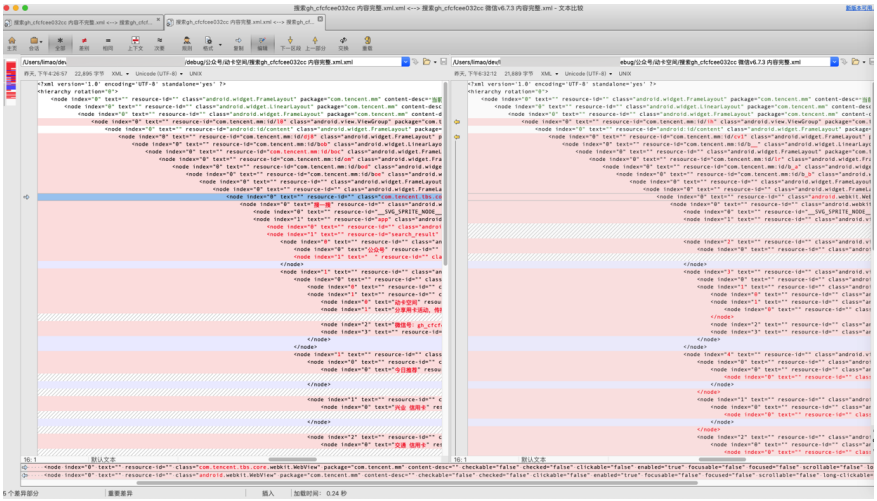
- 升级前 = 微信 `v6.7.3`
  - 2018微信v6.7.3老旧历史版本安装包官方免费下载_豌豆荚
  - 页面源码：

```
<node index="0" text="" resource-id="" class="and
<node index="0" text="" resource-id="" class="and

<node index="0" text="" resource-id="__SVG_SPRITE
...
```

- 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：

```
<node index="0" text="" resource-id="" class="com
<node index="0" text="搜一搜" resource-id="" class

<node index="0" text="" resource-id="__SVG_SPRITE
...
```

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）都可能会被判定为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、表格等方面的节点，
        - 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

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：

```
uiautomator2.exceptions.JsonRpcError: 0 Unknown error: <In
```

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

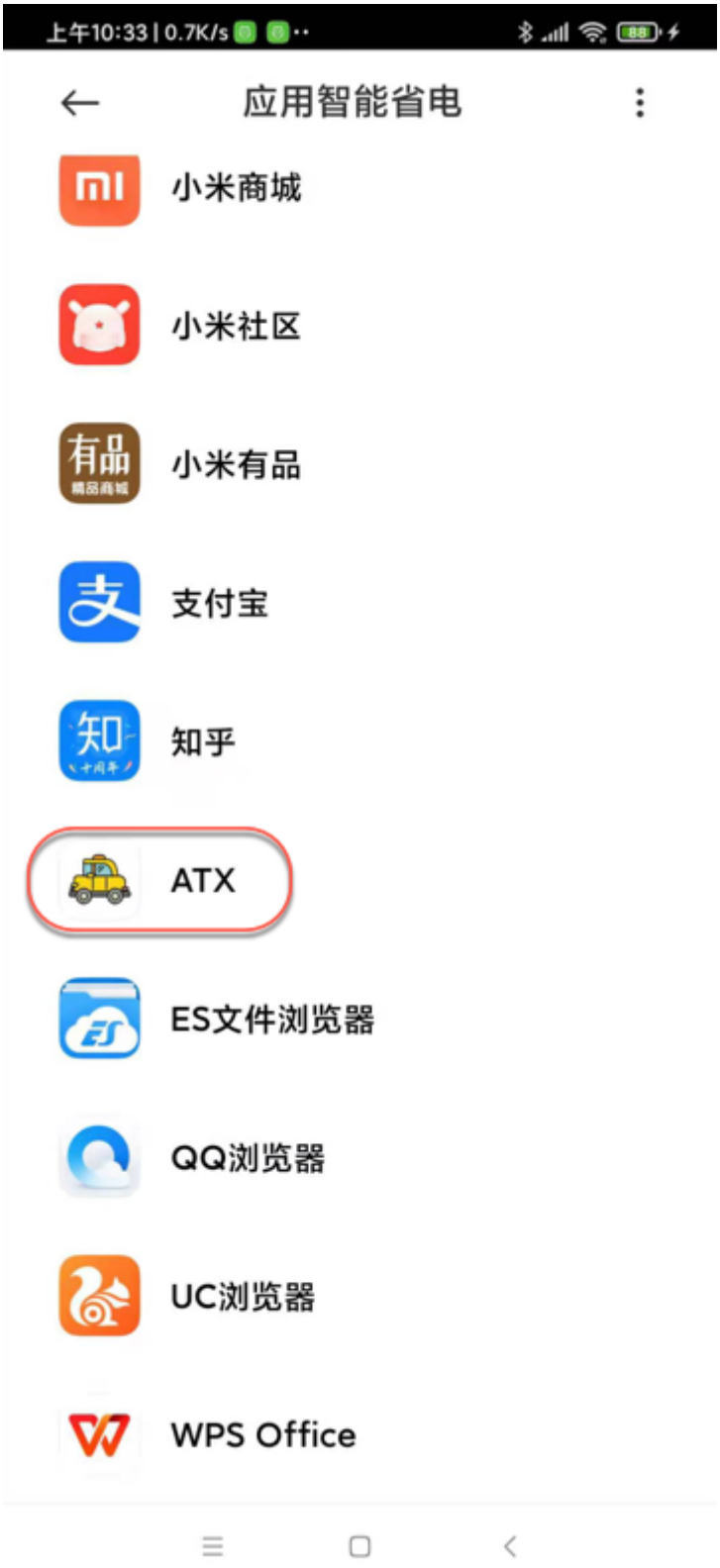
- 自己此处是调试手机，自动抓包，所以没问题，是打算开启此权限

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


上午10:33 | 0.7K/s
应用智能省电
小米商城
小米社区
小米有品
支付宝
知乎
ATX
ES文件浏览器
QQ浏览器
UC浏览器
WPS Office

- com.github.uiautomator.test


上午10:36 | 0.8K/s
应用智能省电
小米商城
小米社区
小米有品
支付宝
知乎
com.github.uiautomator.test
ES文件浏览器
QQ浏览器
UC浏览器
WPS Office

。

其他还有类似的：

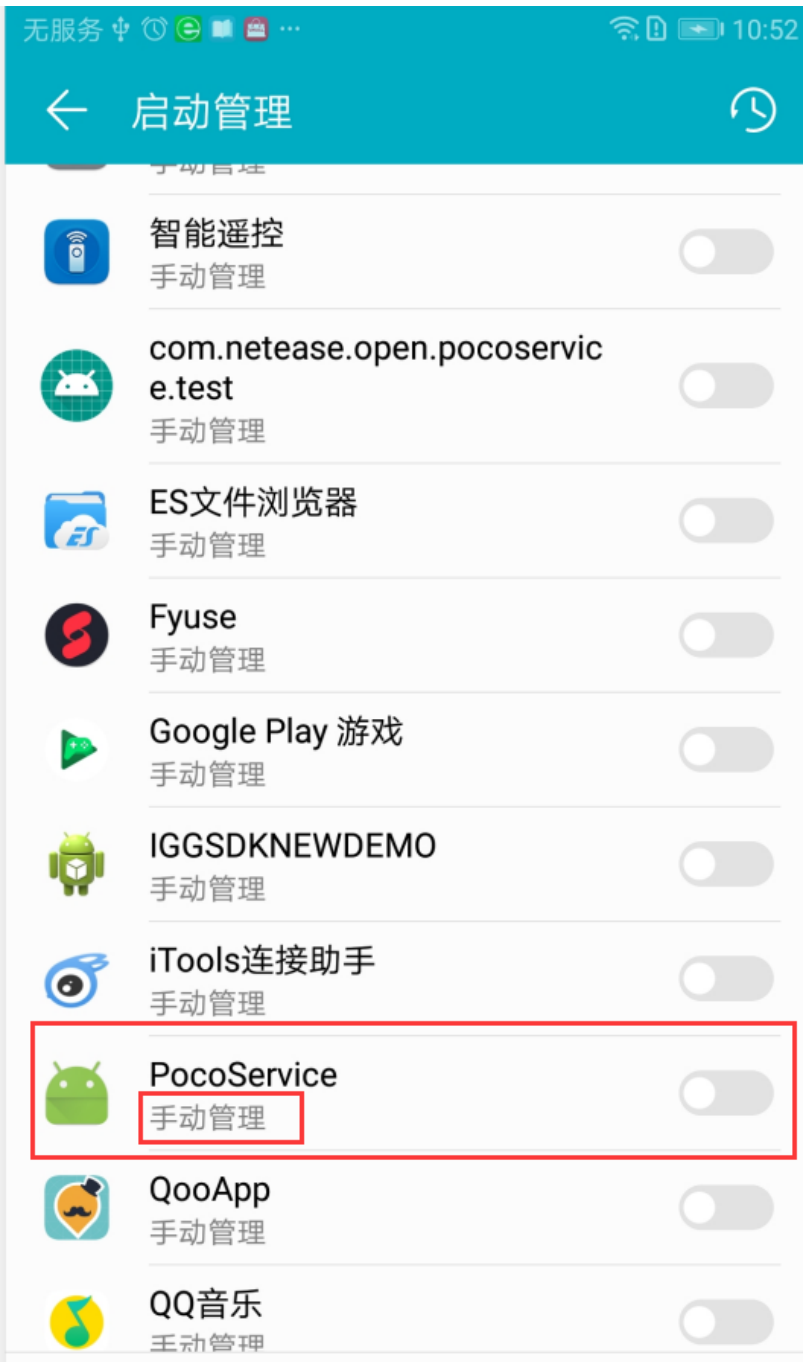
在设置中的 `后台高耗电` 中：

- 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.googlecode.jsonrpc4j.JsonRpcServer;

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



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

其中对于launchService：

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


        // Wait for launcher
        String launcherPackage = device.getLauncherPackageN
        Boolean ready = device.wait(Until.hasObject(By.pkg(
        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.
        intent.setPackage("com.github.uiautomator"); // fix
        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 AutomatorServi

    /**
     * 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 e
public boolean swipe(int startX, int startY, int endX, int
...
```

都是常见的基础功能。

```
    // Multi touch is a little complicated
    @Override
    public boolean injectInputEvent(int action, float x, f
        MotionEvent e = MotionEvent.obtain(SystemClock.upt
                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, r
     * @return true if successful, else return false
     * @throws RemoteException
     */
    @Override
    public boolean pressKey(String key) throws RemoteExcep
        boolean result;
        key = key.toLowerCase();
        if ("home".equals(key)) result = device.pressHome(
        else if ("back".equals(key)) result = device.pressB
        else if ("left".equals(key)) result = device.pressD
        else if ("right".equals(key)) result = device.press
        else if ("up".equals(key)) result = device.pressDPa
        else if ("down".equals(key)) result = device.pressD
        else if ("center".equals(key)) result = device.pres
        else if ("menu".equals(key)) result = device.pressM
        else if ("search".equals(key)) result = device.pres
        else if ("enter".equals(key)) result = device.press
        else if ("delete".equals(key) || "del".equals(key))
        else if ("recent".equals(key)) result = device.pres
        else if ("volume_up".equals(key)) result = device.p
        else if ("volume_down".equals(key))
            result = device.pressKeyCode(KeyEvent.KEYCODE_V
        else if ("volume_mute".equals(key))
            result = device.pressKeyCode(KeyEvent.KEYCODE_V
        else if ("camera".equals(key)) result = device.pres
        else result = "power".equals(key) && device.pressKe


        return result;
    }



public boolean pressKeyCode(int keyCode) {
public boolean pressKeyCode(int keyCode, int metaState) {

public void clearTextField(Selector obj) throws UiObjectNot




    /**
     * Reads the text property of the UI element
     *
     * @param obj the selector of the UiObject.
     * @return text value of the current node represented b
```

```
     * @throws UiObjectNotFoundException
     */
    @Override
    public String getText(Selector obj) throws UiObjectNotF
        if (obj.toUiObject2() == null) {
            return device.findObject(obj.toUiSelector()).ge
        } else {
            return obj.toUiObject2().getText();
        }
    }


    /**
     * Sets the text in an editable field, after clearing t
     *
     * @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) throw
        try {
            obj.toUiObject2().click();
            obj.toUiObject2().setText(text);
            return true;
        } catch (NullPointerException | StaleObjectExceptio
            return device.findObject(obj.toUiSelector()).se
        }
    }


    /**
     * Performs a click at the center of the visible bounds
     *
     * @param obj the target ui object.
     * @return true id successful else false
     * @throws UiObjectNotFoundException
     */
    @Override
    public boolean click(Selector obj) throws UiObjectNotFo
        if (obj.toUiObject2() == null) {
            return device.findObject(obj.toUiSelector()).cl
        } else {
            obj.toUiObject2().click();
            return true;
        }
    }


    /**
```

```
     * Clicks the bottom and right corner or top and left c
     *
     * @param obj    the target ui object.
     * @param corner "br"/"bottomright" means BottomRight,
     * @return true on success
     * @throws UiObjectNotFoundException
     */
    @Override
    public boolean click(Selector obj, String corner) throw
        return click(device.findObject(obj.toUiSelector()),
    }


    private boolean click(UiObject obj, String corner) thro
        if (corner == null) corner = "center";
        corner = corner.toLowerCase();
        if ("br".equals(corner) || "bottomright".equals(cor
        else if ("tl".equals(corner) || "topleft".equals(co
        else if ("c".equals(corner) || "center".equals(corn
        return false;
    }


public boolean dragTo(Selector obj, Selector destObj, int s




    /**
     * Performs the swipe up/down/left/right action on the
     *
     * @param obj   the target ui object.
     * @param dir   "u"/"up", "d"/"down", "l"/"left", "r"/"
     * @param steps indicates the number of injected move s
     * @return true of successful
     * @throws UiObjectNotFoundException
     */
    @Override
    public boolean swipe(Selector obj, String dir, int step
        return swipe(device.findObject(obj.toUiSelector()),
    }


    private boolean swipe(UiObject item, String dir, int st
        dir = dir.toLowerCase();
        boolean result = false;
        if ("u".equals(dir) || "up".equals(dir)) result = t
        else if ("d".equals(dir) || "down".equals(dir)) res
        else if ("l".equals(dir) || "left".equals(dir)) res
        else if ("r".equals(dir) || "right".equals(dir)) re
        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
     *
     * @param compressed use compressed layout hierarchy or
     * @param filename   the filename to be stored. @deprec
     * @return the absolute path name of dumped file.
     */
    @Deprecated
    @Override
    public String dumpWindowHierarchy(boolean compressed, S
        return dumpWindowHierarchy(compressed);
    }

    /**
     * Helper method used for debugging to dump the current
     *
     * @param compressed use compressed layout hierarchy or
     * @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.dumpWindowHierarchy(os);


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


        return null;
    }
```

前一个：

```
public String dumpWindowHierarchy(boolean compressed, Strin
```

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,
        XmlSerializer serializer = Xml.newSerializer();
        serializer.setFeature("http://xmlpull.org/v1/doc/fe
        serializer.setOutput(out, "UTF-8");
        serializer.startDocument("UTF-8", true);
        serializer.startTag("", "hierarchy");
        serializer.attribute("", "rotation", Integer.toStri
        AccessibilityNodeInfo[] arr$ = getWindowRoots(devic
        int len$ = arr$.length;


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


        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_u
```

底层发送的请求是：

```
Shell$ curl -X POST -d 'b'{"jsonrpc": "2.0", "id": "5a175f3
```

最终返回的结果是：

```
Output> {"jsonrpc":"2.0","id":"5a175f3159cc1aa2e27f1cb68f5a
```

可见其中的xml头部的内容：

```
<?xml version='1.0' encoding='UTF-8' standalone='yes' ?><h
```

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

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。

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, sh
        try:
            resultStr = consoleOutputByte.decode("utf-8")
        except UnicodeDecodeError:
            # TODO: use chardet auto detect encoding
            # consoleOutputStr = consoleOutputByte.decode('
            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
```

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'.forr
    cmd = 'adb -s {} shell getprop ro.product.name'.format(
    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 r
```

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`, F
    Returns:
        device list(list)
    Raises:
    Examples:
        output:
            False -> ['2e2a0cb1', 'orga4pmzee4ts47t', '192.
            True -> [{'2e2a0cb1': {'usb': '338952192X', 'pr
    """
    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', 'org
    """
    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 produc
        orga4pmzee4ts47t       device usb:338886656X produc
        192.168.31.84:5555     device product:PD2065 model:
    """

    for eachLine in deviceLines:
        if not eachLine:
            continue

        if "devices attached" in eachLine:
            continue

        foundDevice = re.search("(?P<devSerial>[\w\.\:]+)\s
        logging.debug("foundDevice=%s", foundDevice)
        # foundDevice=<re.Match object; span=(0, 101), matc
        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 r
                keyValueIter = re.finditer("(?P<key>\w+):(?
                keyValueMatchList = list(keyValueIter)
                logging.debug("keyValueMatchList=%s", keyVa
                # keyValueMatchList=[<re.Match object; span
                detailInfoDict = {}
                for eachMatch in keyValueMatchList:
                    eachKey = eachMatch.group("key")
                    eachValue = eachMatch.group("value")
                    detailInfoDict[eachKey] = eachValue
                logging.debug("detailInfoDict=%s", detailIn
                # detailInfoDict={'usb': '338952192X', 'pro
                curDevDetailDict = {
                    devSerial: detailInfoDict
                }
                logging.debug("curDevDetailDict=%s", curDev
                # curDevDetailDict={'2e2a0cb1': {'usb': '33
                deviceList.append(curDevDetailDict)
            else:
                deviceList.append(devSerial)

    logging.debug("deviceList=%s", deviceList)
    # deviceList=[{'2e2a0cb1': {'usb': '338952192X', 'produ
    # ['2e2a0cb1', 'orga4pmzee4ts47t', '192.168.31.84:5555
    return deviceList
```

调用：

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

或：

```
deviceDetailList = self.getAndroidDeviceList(isGetDetail=Tr
# [{'2e2a0cb1': {'usb': '338952192X', 'product': 'PD2065',
```

检测安卓设备是否连接

```
def isAndroidUsbConnected(self, deviceSerialId):
    """Check whether android device is currently USB wired

    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: 192.168.31.84:555
        logging.error("Invalid android USB wired connected
        return isUsbConnected

    deviceDetailList = self.getAndroidDeviceList(isGetDeta
    for eachDevDetailDict in deviceDetailList:
        curDevSerialStr, curDevDetailDict = list(eachDevDe
        if deviceSerialId == curDevSerialStr:
            detailInfoKeyList = list(curDevDetailDict.keys
            # ['usb', 'product', 'model', 'device', 'transp
            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 d

    Args:
        wifiSerial (str): android devivce WiFi serial, eg:
    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", adbConne
    # os.system(adbConnectCmd) # when failed, will wait too
    cmdOutputStr = CommonUtils.get_cmd_lines(adbConnectCmd,
    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
    # "failed to connect to '192.168.31.84:5555': Connectio
    # err=Command 'adb connect 192.168.31.84:5555' timed ou
    if cmdOutputStr:
        if "connected" in cmdOutputStr:
            isConnectOk = True
        elif ("failed" in cmdOutputStr) or ("timed out" in
            isConnectOk = False
    else:
        isConnectOk = False

    return isConnectOk
```

调用：

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

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 = "",""
    cmds = ['dumpsys activity |grep {}'.format(item) for it
    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 = 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(se
        lines = CommonUtils.get_cmd_lines(cmd)
        return {line.split(":")[1].strip() for line in line
```

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 alread installed".forr
    else:
        logging.info("start to install app in {}".format(os
        os.system("adb -s {0} install {1}".format(self.devi
```

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 {0} uninstall {1}".format(self.device
    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(dev
    time.sleep(1)
    # cmds = [
    #     'adb -s {} shell dumpsys window policy | grep isS
    #     'adb -s {} shell dumpsys window policy | grep mSh
    #     'adb -s {} shell dumpsys window policy | grep mDr
    # ]
    # for cmd in cmds:
    #     text = CommonUtils.get_cmd_lines(cmd,text=True)
    #     if text and "=true" in text:
    #         logging.info("start to unlock device {}".forr
    #         return False

    # cmds = [
    #     'adb -s {} shell dumpsys power | grep mHoldingDis
    # ]
    # for cmd in cmds:
    #     text = CommonUtils.get_cmd_lines(cmd,text=True)
    #     if text and "=false" in text:
    #         # '  mHoldingDisplaySuspendBlocker=false\n'
    #         logging.info("start to unlock device {}".forr
    #         return False

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

    logging.info("device {} is already unlock".format(devic
    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,she
    # level = self.driver.shell(shell_cmd).output if isinst
    level = self.driver.shell(shell_cmd).output if isinstan
    result = re.search("\d+",level)
    ratio = int(result.group()) if result else 100
    return ratio
```

调用：

```
batteryElectricityPercentInt = self.get_device_electricity_
```

获取安卓手机名

```
def get_phone_name_Android(self):
    # cmd = 'adb -s {} shell getprop ro.product.model'.forr
    cmd = 'adb -s {} shell getprop ro.product.name'.format(
    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()
```

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',
```

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', 'c
```

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_RESTA
        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", dis
    # 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", screer
    if driverInfo["displayRotation"]:
        curScreenWidth = screenHeight
        curScreenHeight = screenWidth
    else:
        curScreenWidth = screenWidth
        curScreenHeight = screenHeight
    logging.debug("curScreenWidth=%s, curScreenHeight=%s",

    return (curScreenWidth, curScreenHeight)
```

调用：

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

输出：

```
[191213 16:16:13][AppCrawler.py 209] driverInfo={'currentPa
[191213 16:16:13][AppCrawler.py 212] displayWidth=1196, dis
[191213 16:16:13][AppCrawler.py 214] deviceInfo={'udid': 'D
[191213 16:16:13][AppCrawler.py 216] deviceDisplay={'width
[191213 16:16:13][AppCrawler.py 219] screenWidth=720, scree
[191213 16:16:13][AppCrawler.py 226] curScreenWidth=1280, c
```

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

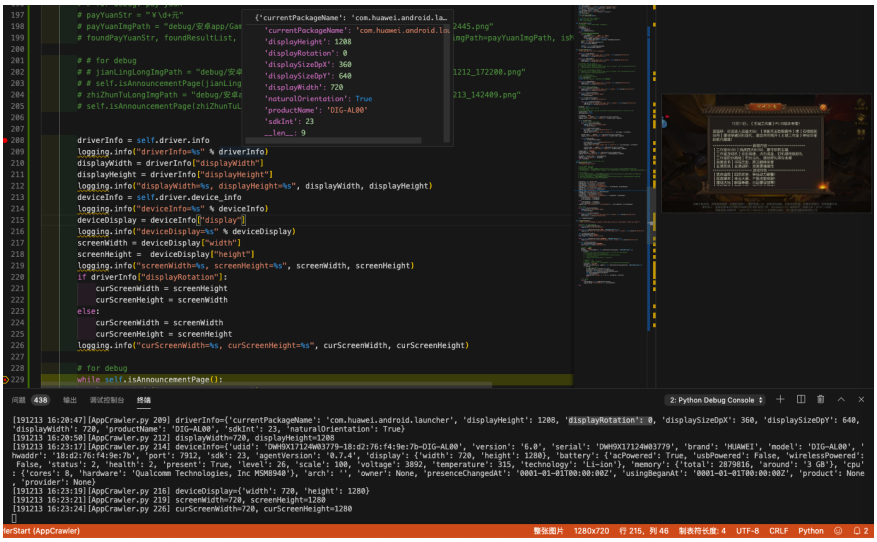
另外，当屏幕故意不去旋转，回到默认竖屏后：


USB 3.0
HDTV
USB 3.0
USB-C
USB 3.0
16:21
Bach Suite
热点
图库
文件管理
主题
QQ浏览器
应用市场
信息
设置
相机
HUAWEI

此时

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

如图：



详细log：

```
[191213 16:20:47][AppCrawler.py 209] driverInfo={'currentPa
[191213 16:20:50][AppCrawler.py 212] displayWidth=720, disp
[191213 16:23:17][AppCrawler.py 214] deviceInfo={'udid': 'D
[191213 16:23:19][AppCrawler.py 216] deviceDisplay={'width
[191213 16:23:21][AppCrawler.py 219] screenWidth=720, scree
[191213 16:23:24][AppCrawler.py 226] curScreenWidth=720, cu
```

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

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.
        bounds = element.attrib.get("bounds")
        return list(map(int, re.sub('\[|,|\]', " ", bounds)
    elif self.isiOS:
        # <XCUIElementTypeButton 。。。 name="返回" label="返
        # 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

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(elemer
    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, "B
    elif self.isiOS:
        # <XCUIElementTypeButton type="XCUIElementTypeButto
        # iOSTagButton = "XCUIElementTypeButton"
        # elementTag = element.tag
        # isButton = elementTag == iOSTagButton
        # return isButton
        return self.is_element_SomeType_iOS(element, "XCUIE

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, "I
    elif self.isiOS:
        # <XCUIElementTypeImage type="XCUIElementTypeImage"
        # iOSTagImage = "XCUIElementTypeImage"
        # elementTag = element.tag
        # isImage = elementTag == iOSTagImage
        # return isImage
        return self.is_element_SomeType_iOS(element, "XCUIE

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, "E
    elif self.isiOS:
        # <XCUIElementTypeStaticText type="XCUIElementTypeS
        # iOSTagStaticText = "XCUIElementTypeStaticText"
        # elementTag = element.tag
        # isStaticText = elementTag == iOSTagStaticText
        # return isStaticText
        # return self.is_element_SomeType_iOS(element, "XCU

        # <XCUIElementTypeSearchField type="XCUIElementType
        #    <XCUIElementTypeButton type="XCUIElementTypeBu
        # </XCUIElementTypeSearchField>
        isSearchField = self.is_element_SomeType_iOS(elemen
        # <XCUIElementTypeTextField type="XCUIElementTypeTe
        isTextField = self.is_element_SomeType_iOS(element,
        #
        isSecureTextField = self.is_element_SomeType_iOS(el
        isEditableText = isSearchField or isTextField or is
        return isEditableText
```

```
def is_element_Link(self, element):
    # 元素是否是 XCUIElementTypeLink
    """
        <XCUIElementTypeLink type="XCUIElementTypeLink" nar
            <XCUIElementTypeStaticText type="XCUIElementTyp
        </XCUIElementTypeLink>
    """
    return self.is_element_SomeType_iOS(element, "XCUIEleme

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
    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 a
            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", centerP
            hasClicked = True

        return hasClicked
```

调用：

```
moreInfoSoup = parentCellSoup.find(
    'XCUIElementTypeButton',
    attrs={"type": "XCUIElementTypeButton", "name": "更多信息
)
if moreInfoSoup:
    clickedOk = self.clickElementCenterPosition(moreInfoSou
```

或：

```
page = self.get_page_source()
backElement, newPage = self.findRealBackElement(page)
if backElement is not None:
    isFoundAndClicked = self.clickElementCenterPosition(bac
```

或：

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

或：

```
isGetProxyTypeOk, respInfo = self.iOSLaunchSettingsAndGetPr
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.osIsWinows():
        # Windows
        cmd = "wmic bios get serialnumber"
    elif CommonUtils.osIsMacOS():
        # macOS
        cmd = "system_profiler SPHardwareDataType | awk '/S
    # TODO: add support other OS
    # AIX: aix
    # Linux: linux
    # Windows/Cygwin: cygwin

    serialNumber = ""
    lines = CommonUtils.get_cmd_lines(cmd)
    if CommonUtils.osIsWinows():
        # Windows
        serialNumber = lines[1]
    elif CommonUtils.osIsMacOS():
        # macOS
        serialNumber = lines[0] # C02Y3N10JHC8, 'VMfvNykazW

    return serialNumber
```

调用：

```
serialNumber = self.getSerialNumber() # 'VMfvNykazWi1'
```

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 sessio
    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=scaledSi
    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() # '202
    # suffix = "png"
    suffix = "jpg" # '20200422_144915.jpg'
    curFilename = "%s.%s" % (curDatetimeStr, suffix)
    if not saveFolder:
        if self.isAndroid:
            # saveFolder = self.config["CurAndroidAppScreen
            # saveFolder = self.config["CurAndroidWeixinSc
            # saveFolder = self.config["debug"]["screenshot
            saveFolder = self.config["debug"]["screenshot"]
        elif self.isiOS:
            # saveFolder = self.config["CuriOSWeixinScreens
            # saveFolder = self.config["CuriOSAppPageSource
            # saveFolder = self.config["debug"]["pageSource
            # saveFolder = self.config["debug"]["screenshot
            # saveFolder = self.config["debug"]["screenshot
            saveFolder = self.config["debug"]["screenshot"]
    # add current date sub folder
    curDateStr = CommonUtils.getCurDatetimeStr("%Y%m%d") #
    saveFolder = os.path.join(saveFolder, curDateStr) # 'de
    CommonUtils.createFolder(saveFolder)
    fullImgFilePath = os.path.join(saveFolder, curFilename)
    beforeDriverSceenshotTime = datetime.now()
    if self.isAndroid:
        fullImgFilePath = self.driver.screenshot(fullImgFil
        # optimize size
        displayInfo = self.driver.device_info["display"] #
        originSize = (displayInfo["height"], displayInfo["w
        CommonUtils.resizeImage(fullImgFilePath, originSize
    elif self.isiOS:
        fullImgFilePath = self.debugiOSSaveScreenshot(saveF
    afterDriverSceenshotTime = datetime.now()
    driverSceenshotTime = afterDriverSceenshotTime - before
    logging.debug("driver screenshot time: %s", driverScree
    return fullImgFilePath
```

给当前屏幕截图加标记（红框）

```
def debugDrawScreenRect(self, curRect, curImgPath=None, isS
    """for debug, draw rectange for current screenshot"""
    if not curImgPath:
        curImgPath = self.getCurScreenshot()

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

    return curImg
```

给元素加边框标记

```
def debugDrawElementRect(self, elementList, curImgPath=None
    """for debug, to draw rectange for each element in curr
    if not curImgPath:
        curImgPath = self.getCurScreenshot()

    curImg = Image.open(curImgPath)

    for eachElement in elementList:
        curBoundList = self.get_ElementBounds(eachElement)
        curWidth = curBoundList[2] - curBoundList[0]
        curHeight = curBoundList[3] - curBoundList[1]
        curRect = [curBoundList[0], curBoundList[1], curWid
        curTimeStr = CommonUtils.getCurDatetimeStr("%H%M%S"
        curSaveTal = "_rect_{}_%x|%y|%w|%h".format(curTimeS
        curInputImg = None
        if isDrawInSinglePic:
            curInputImg = curImg
        else:
            curInputImg = curImgPath
        curImg = CommonUtils.imageDrawRectangle(
            curInputImg,
            curRect,
            isShow=isShowEach,
            isAutoSave=isSaveEach,
            saveTail=curSaveTal,
            isDrawClickedPosCircle=False,
        )

    # always save final result
    curTimeStr = CommonUtils.getCurDatetimeStr("%H%M%S")
    finalSaveTail = "_rect_all_%s" % curTimeStr
    imgFolderAndName, pointSuffix = os.path.splitext(curImg
    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=
    """for debug, save current page source xml file"""
    savedSourceFile = None
    curDatetimeStr = CommonUtils.getCurDatetimeStr()
    sourceFormat="xml"
    # sourceFilename = "%s_source.%s" % (curDatetimeStr, so
    sourceFilename = "%s.%s" % (curDatetimeStr, sourceForma
    if filePrefix:
        sourceFilename = "%s_%s" % (filePrefix, sourceFiler
        # 'com.netease.cloudmusic_20200221_170337.xml'

    if not saveFolder:
        # if self.isAndroid:
        #     # saveFolder = self.config["CurAndroidAppPage
        #     # saveFolder = self.config["CurAndroidWeixinP
        #     # saveFolder = self.config["debug"]["pageSour
        #     saveFolder = self.config["debug"]["pageSource
        # elif self.isiOS:
        #     # saveFolder = self.config["CuriOSWeixinPageS
        #     # saveFolder = self.config["debug"]["pageSour
        #     saveFolder = self.config["debug"]["pageSource

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

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

    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", "
        else:
            curInfoDict = curElement.info
            keyList = ["resourceName", "className", "bounds
    elif self.isiOS:
        curInfoDict = curElement.attrib
        # keyList = ["type", "name", "label", "value", "ena
        keyList = ["type", "name", "label", "value", "enab

    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, cu

    curElementStr = " | ".join(valueList)
    logging.info("%s element: %s", prefix, curElementStr)
    return
```

调用：

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

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：

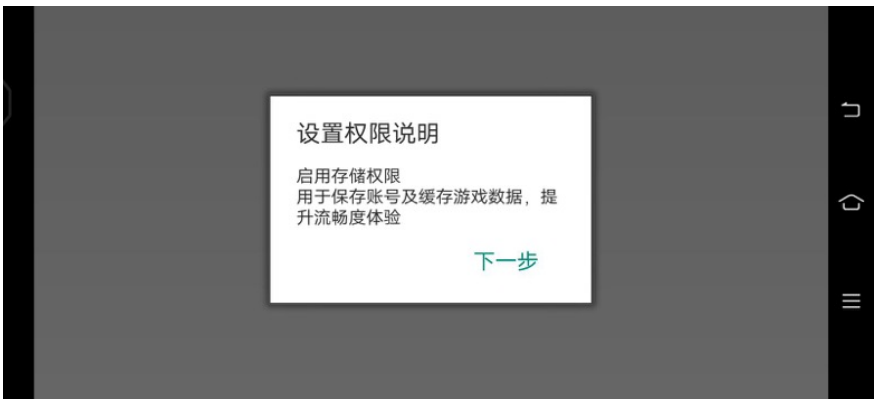
下一步

代码：

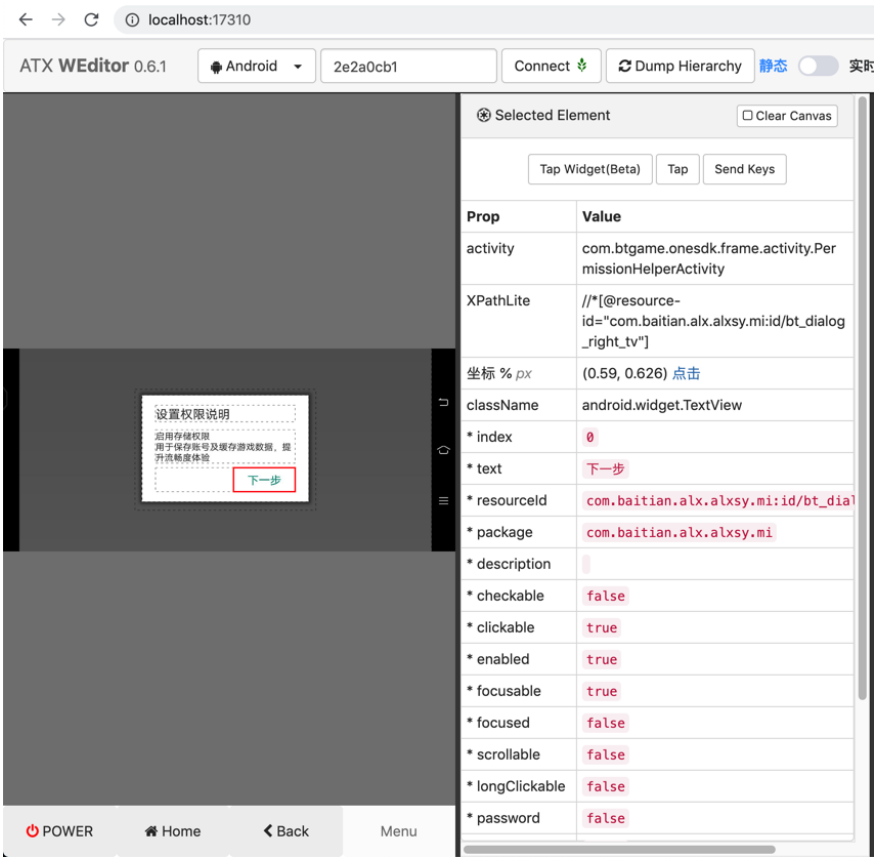
```
NextStep_Button_Xpath_List: [
    "//android.widget.TextView[@text='下一步' and contains(@
]

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.PermissionHelp
XPathLite    //*[@resource-id="com.baitian.alx.alxsy.mi:id/
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
* 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

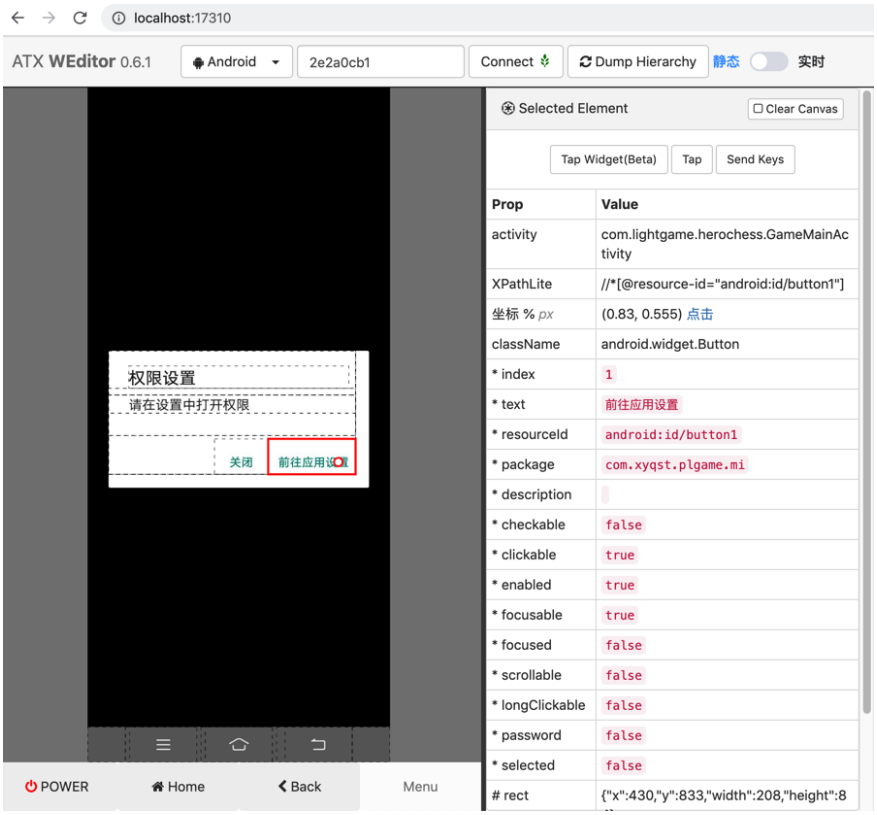
自动同意前往应用设置的权限

代码：

```
Permission_Settings_GoTo_App_Config_Xpath: "//android.widge

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=gotoAppSettin
    if gotoAppSettingsElement:
        gotoAppSettingsElement.click()
        logging.info("Has clicked %s button", gotoAppSettin
        time.sleep(0.1)
    else:
        logging.warning("Not found %s button", gotoAppSett
```

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

微信相关

查找微信公众号中文全名

代码：

```
# def findWeixinPublicAccountZhcnSoup(self, soup, curAccou
def findWeixinPublicAccountZhcnFullName(self, soup, curAcc
    """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="XCUIElementTypeOthe
                <XCUIElementTypeOther type="XCUIElementType
                    <XCUIElementTypeOther type="XCUIElement
                        <XCUIElementTypeStaticText type="XC
                    </XCUIElementTypeOther>
                </XCUIElementTypeOther>
                <XCUIElementTypeOther type="XCUIElementType
                <XCUIElementTypeOther type="XCUIElementType
                    <XCUIElementTypeStaticText type="XCUIE
                </XCUIElementTypeOther>
                <XCUIElementTypeOther type="XCUIElementType
                    <XCUIElementTypeStaticText type="XCUIE
                </XCUIElementTypeOther>
                <XCUIElementTypeOther type="XCUIElementType
                    <XCUIElementTypeStaticText type="XCUIE
                    <XCUIElementTypeStaticText type="XCUIE
                </XCUIElementTypeOther>

        搜索结果中文名节点是Other，其下是多个Text节点:
            <XCUIElementTypeOther type="XCUIElementTypeOthe
                <XCUIElementTypeOther type="XCUIElementType
                    <XCUIElementTypeStaticText type="XCUIE
                </XCUIElementTypeOther>
                <XCUIElementTypeOther type="XCUIElementType
                <XCUIElementTypeImage type="XCUIElementType
                <XCUIElementTypeImage type="XCUIElementType
                <XCUIElementTypeOther type="XCUIElementType
                    <XCUIElementTypeStaticText type="XCUIE
                    <XCUIElementTypeStaticText type="XCUIE
                    <XCUIElementTypeStaticText type="XCUIE
                </XCUIElementTypeOther>
                <XCUIElementTypeOther type="XCUIElementType
                    <XCUIElementTypeStaticText type="XCUIE
                </XCUIElementTypeOther>
```

```
                <XCUIElementTypeOther type="XCUIElementType
                    <XCUIElementTypeStaticText type="XCUIE
                    <XCUIElementTypeStaticText type="XCUIE
                </XCUIElementTypeOther>

            公众号中文名全部是绿色的:
                <XCUIElementTypeOther type="XCUIElementTypeOthe
                    <XCUIElementTypeOther type="XCUIElementType
                        <XCUIElementTypeOther type="XCUIElement
                            <XCUIElementTypeStaticText type="XC
                        </XCUIElementTypeOther>
                    </XCUIElementTypeOther>
                    <XCUIElementTypeOther type="XCUIElementType
                    <XCUIElementTypeImage type="XCUIElementType
                    <XCUIElementTypeOther type="XCUIElementType
                        <XCUIElementTypeStaticText type="XCUIE
                    </XCUIElementTypeOther>
                    <XCUIElementTypeOther type="XCUIElementType
                        <XCUIElementTypeStaticText type="XCUIE
                    </XCUIElementTypeOther>
                    <XCUIElementTypeOther type="XCUIElementType
                        <XCUIElementTypeStaticText type="XCUIE
                        <XCUIElementTypeStaticText type="XCUIE
                    </XCUIElementTypeOther>
                    <XCUIElementTypeOther type="XCUIElementType
                    <XCUIElementTypeOther type="XCUIElementType
                        <XCUIElementTypeStaticText type="XCUIE
                    </XCUIElementTypeOther>
                </XCUIElementTypeOther>
            """
            foundAccountId = soup.find(
                'XCUIElementTypeStaticText',
                attrs={"value": curAccountId, "name": curAccountId,
            )
            logging.debug("foundAccountId=%s", foundAccountId)
            # foundAccountId=<XCUIElementTypeStaticText enabled="tr
            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.prev
                    # accountDescNode = idParentPrev
                    # logging.info("accountDescNode=%s", accountDes
                    # if accountDescNode:
                    #     # accountZhcnNode = accountDescNode.prev
                    #     accountZhcnNode = accountDescNode.previou
                    #     logging.info("accountZhcnNode=%s", accoun

                    # # method 2: siblings[-2] of XCUIElementTypeOt
```

```
            # idParentPrevSiblingList = idParent.previous_s

            # accountDescNode = None
            # accountZhcnNode = None

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

            # for eachPrevSiblingNode in idParentPrevSiblir
            #     curNodeName = eachPrevSiblingNode.name
            #     isTypeOtherNode = curNodeName == TypeOthe
            #     if isTypeOtherNode:
            #         typeOtherNodeCurIdx += 1

            #         if AccountDescNodeIdx == typeOtherNod
            #             accountDescNode = eachPrevSiblinç
            #         elif AccountZhcnNodeIdx == typeOtherN
            #             accountZhcnNode = eachPrevSiblinç

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

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

            # if accountZhcnNode:
            #     accountZhcnTextSoup = accountZhcnNode.fir
            #         'XCUIElementTypeStaticText',
            #         attrs={ "type": "XCUIElementTypeStat
            #     )

            # method 3: parent'parent is 搜一搜, direct chi
            idParentParent = idParent.parent
            if idParentParent:
                otherSoupList = idParentParent.find_all(
                    "XCUIElementTypeOther",
                    attrs={"type": "XCUIElementTypeOther",
                    recursive=False,
                )
                if otherSoupList and (len(otherSoupList) >=
                    firstOtherSoup = otherSoupList[0]
                    if firstOtherSoup.attrs["name"] == "公众
                        secondOtherSoup = otherSoupList[1]
                        zhcnNameSoupList = secondOtherSoup.
                            "XCUIElementTypeStaticText",
                            attrs={"type": "XCUIElementType
                        )
                        if zhcnNameSoupList:
```

```
                for eachTextSoup in zhcnNameSou
                    curPartName = eachTextSoup.
                    accountZhcnFullName += curP

                if accountZhcnFullName:
                    secondOtherAttrDict = secon
                    parentX = secondOtherAttrDi
                    parentY = secondOtherAttrDi
                    parentWidth = secondOtherAt
                    parentHeight = secondOtherA
                    parentNodeLocator = {
                        "type": "XCUIElementTyp
                        "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

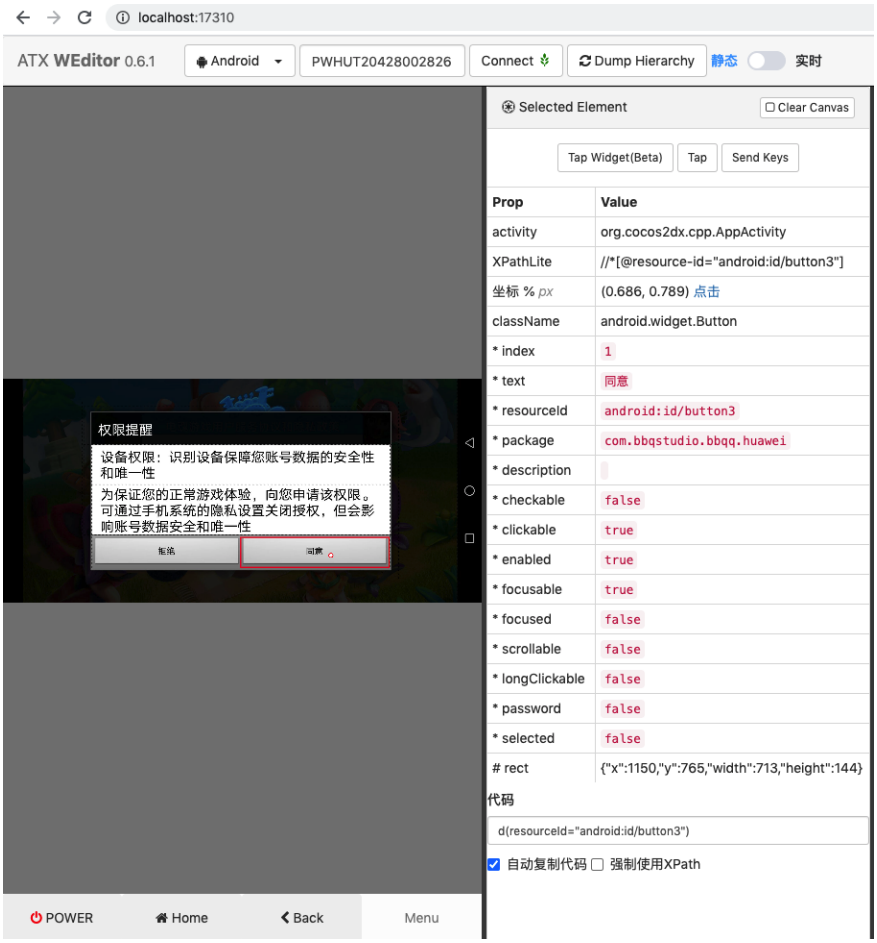
代码：

```
# 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
Confirm_Button_Xpath_List: [
    "//android.widget.Button[@text='确定' and @resource-id=
    "//android.widget.Button[@text='确认' and @resource-id=
    "//android.widget.Button[@text='确定' and @resource-id=
    "//android.widget.Button[@text='确定' and @resource-id=
    "//android.widget.Button[@text='确定' and @resource-id=
    "//android.widget.Button[@text='确定' and contains(@res
    "//android.widget.Button[@text='确定' and @resource-id=
    "//android.widget.Button[@text='同意' and @resource-id=
    "//android.widget.Button[contains(@text, '知道了') and (
    "//android.widget.Button[@text='同意' and contains(@res
    "//android.widget.Button[@text='同意' and @resource-id=
    "//android.widget.LinearLayout[contains(@resource-id,
    # "//android.widget.Button[@text='同意并继续' and @index
    "//android.widget.Button[@text='同意并继续' and @clickab
    "//android.widget.Button[@text='同意继续' and @clickable
    "//android.widget.TextView[@text='同意并继续' and @click
]

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：

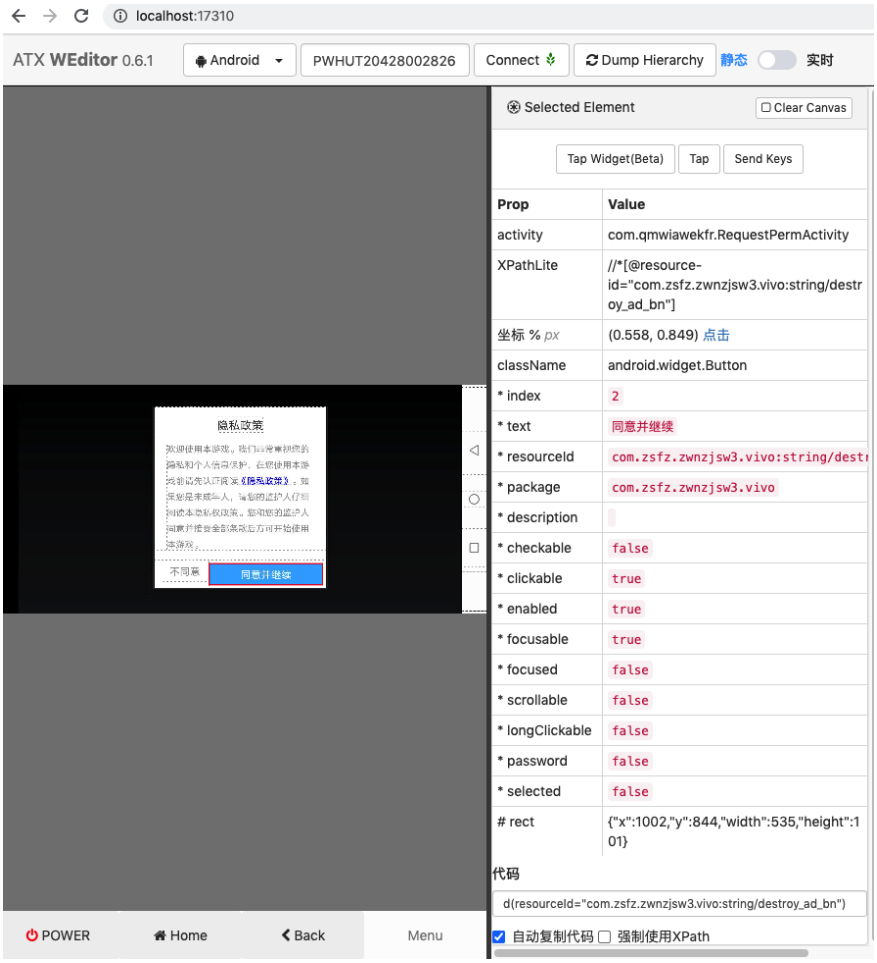


属性：

```
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：

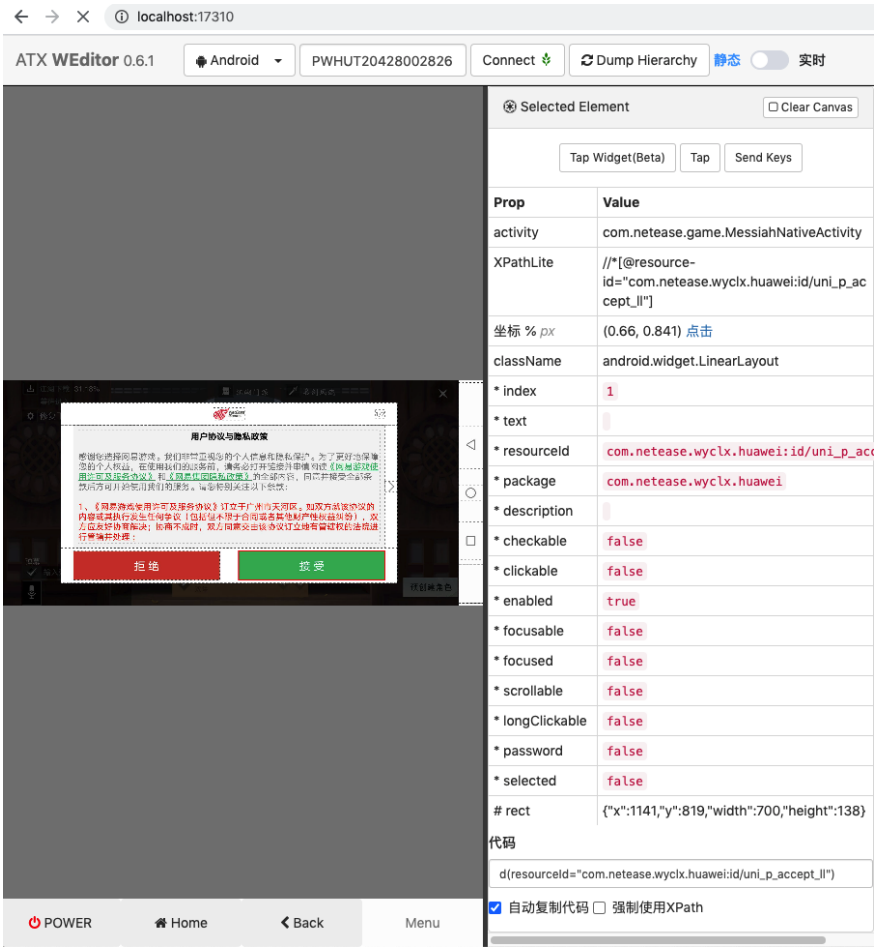


属性：

```
Prop    Value
activity    com.qmwiawekfr.RequestPermActivity
XPathLite    //*[@resource-id="com.zsfz.zwnzjsw3.vivo:strin
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/destroy_ad_br
* 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
```

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：

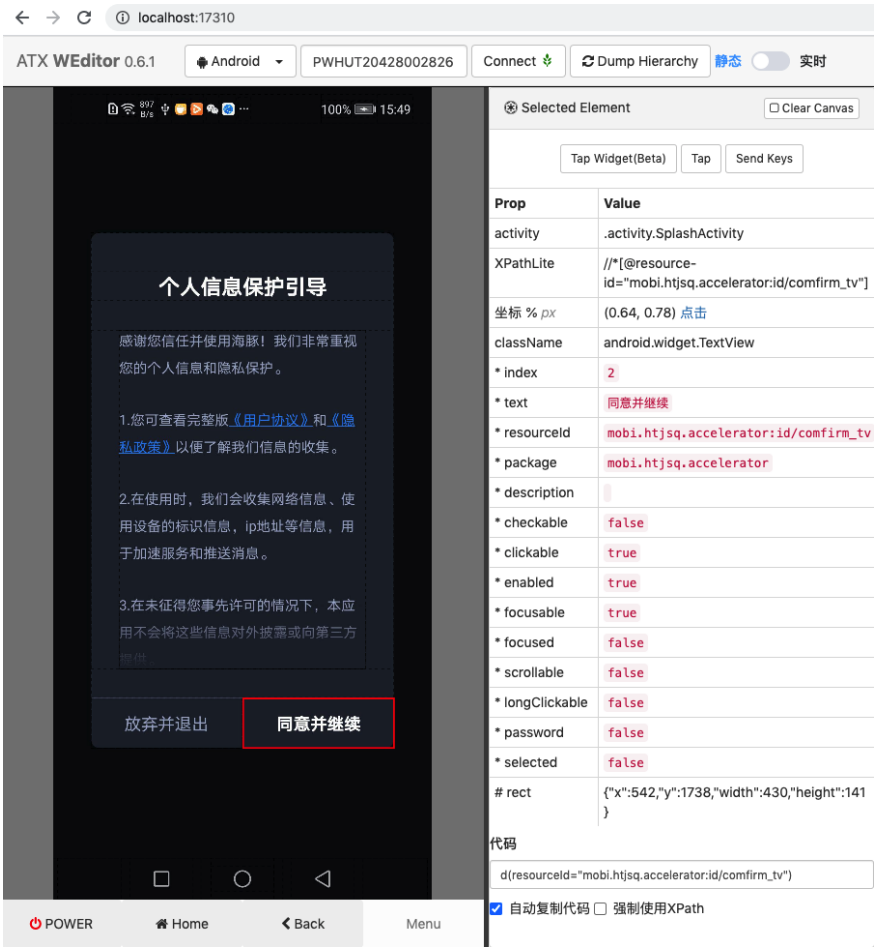


属性：

```
Prop    Value
activity    com.netease.game.MessiahNativeActivity
XPathLite    //*[@resource-id="com.netease.wyclx.huawei:id/
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_l
* 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
```

个人信息保护引导 同意并继续

app截图：

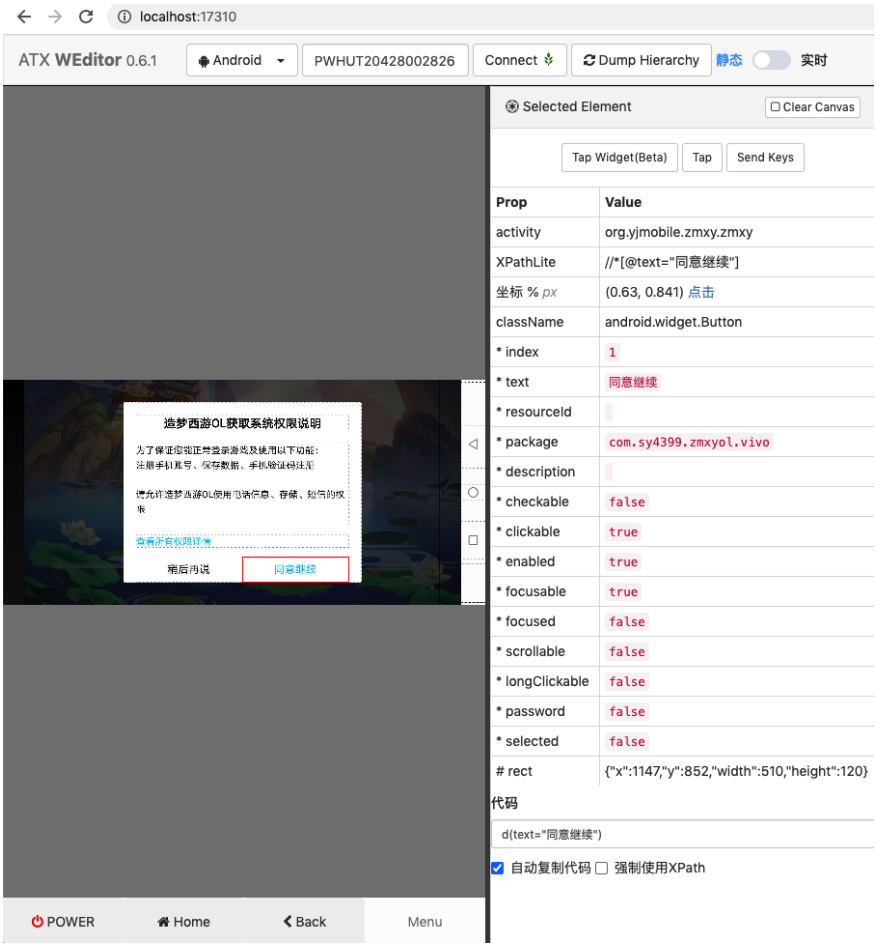


属性：

```
Prop    Value
activity    .activity.SplashActivity
XPathLite    //*[@resource-id="mobi.htjsq.accelerator:id/co
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

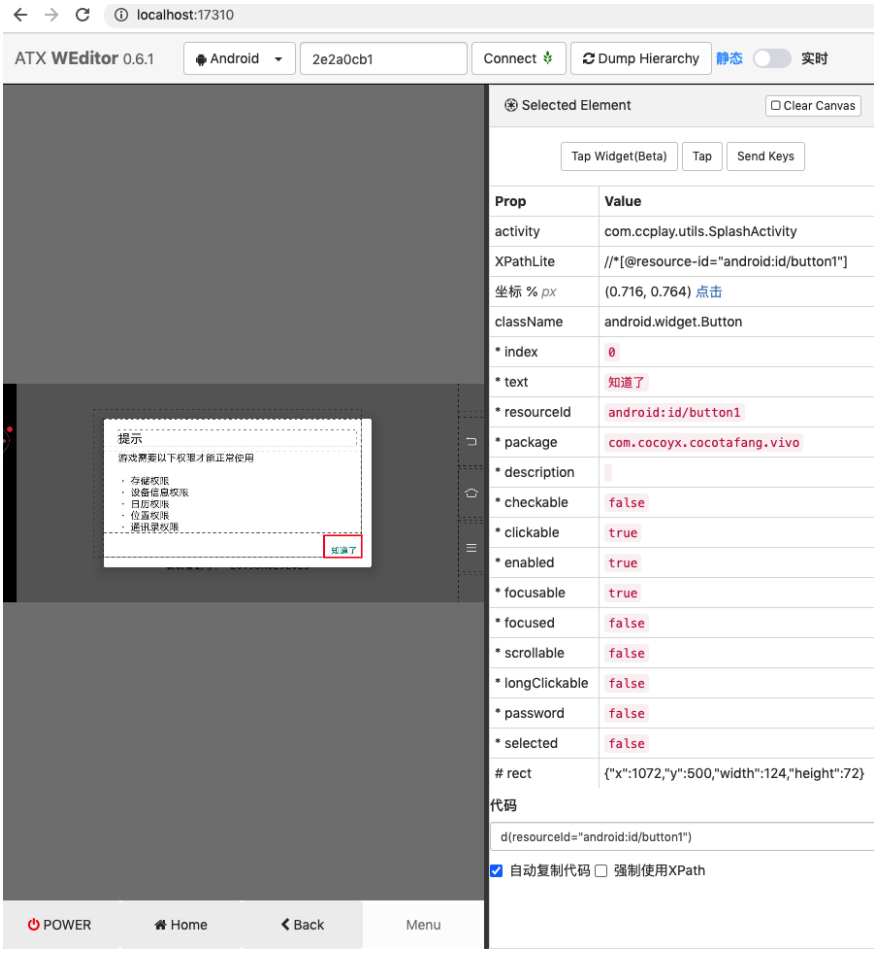
游戏 `com.sy4399.zmxyol.vivo/造梦西游OL-新职业` 的weditor截图:



提示 需要权限 知道了

游戏 `com.cocoyx.cocotafang.vivo/战争模拟器`

weditor截图：

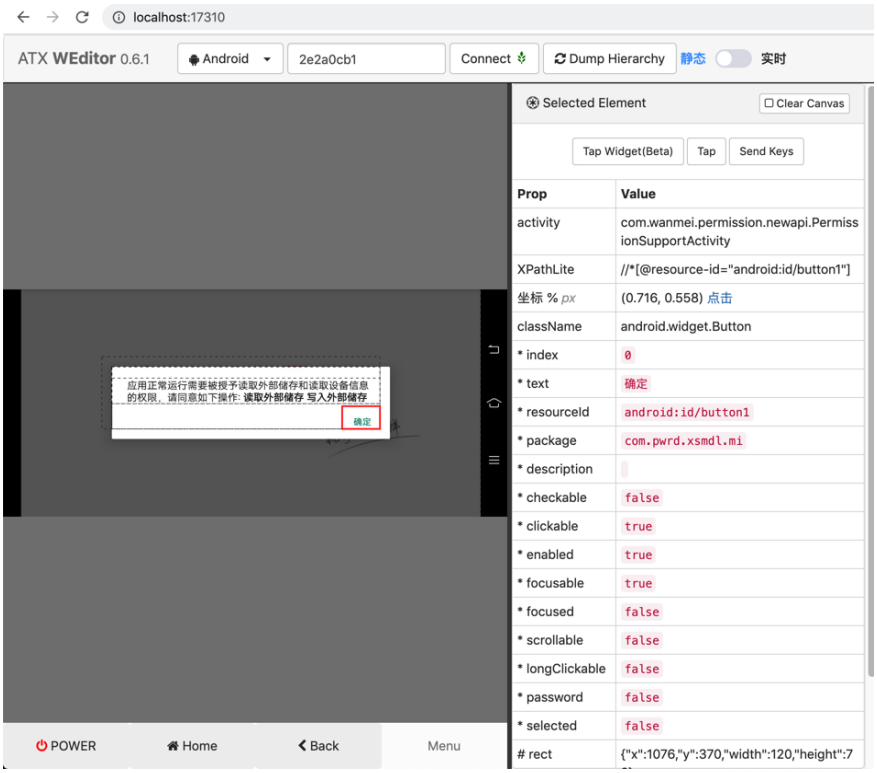


属性：

```
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.cocoyx.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：

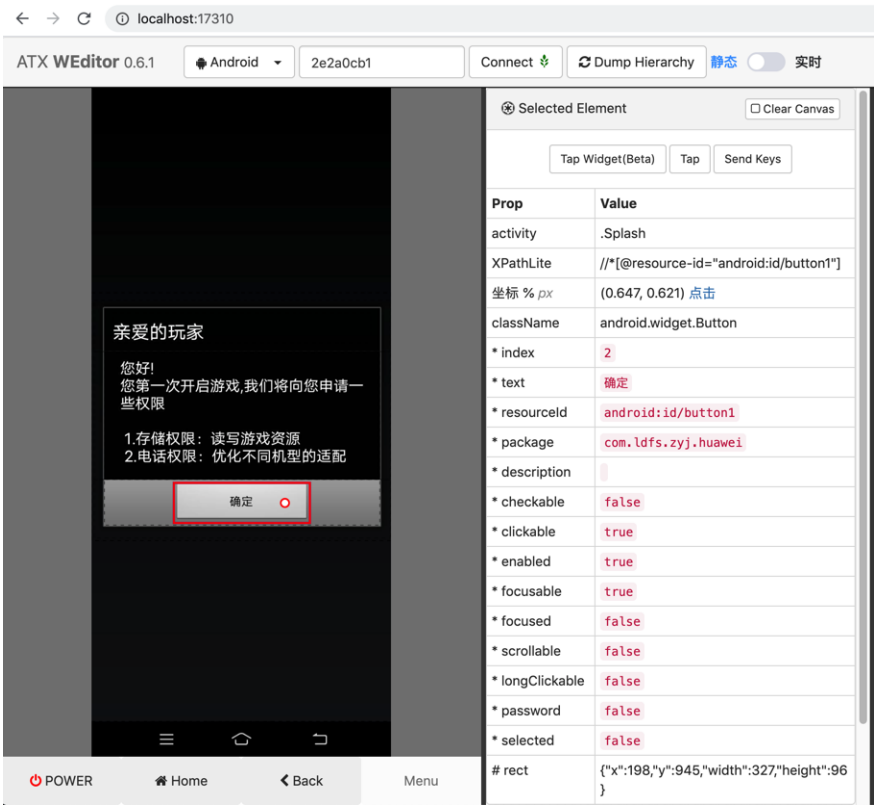


属性：

```
Prop    Value
activity    com.wanmei.permission.newapi.PermissionSupportA
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：



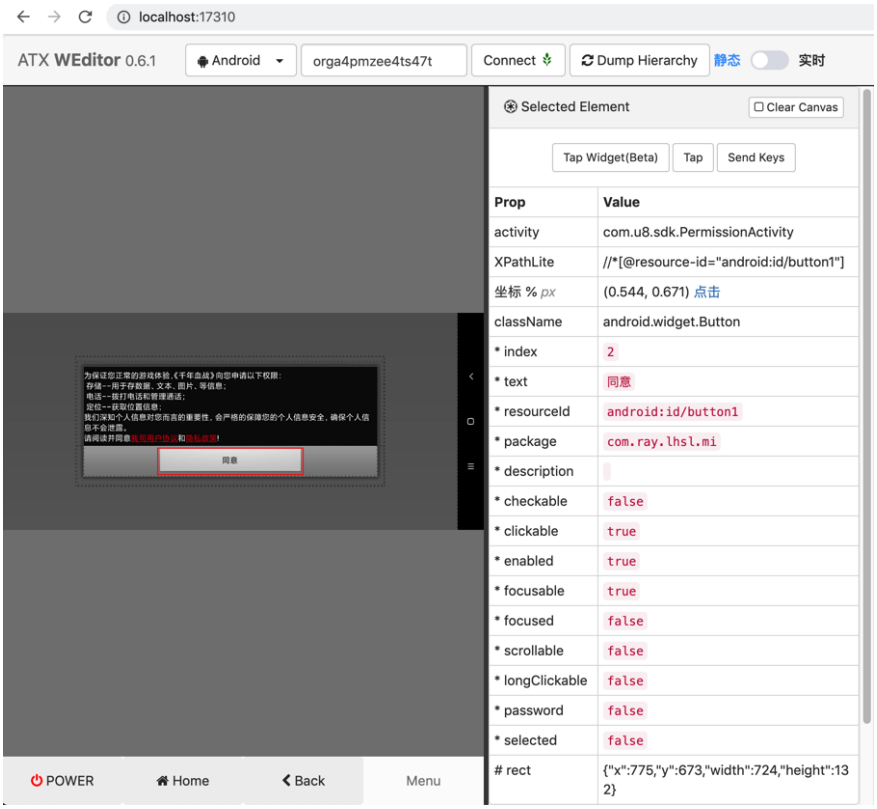
属性：

```
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：

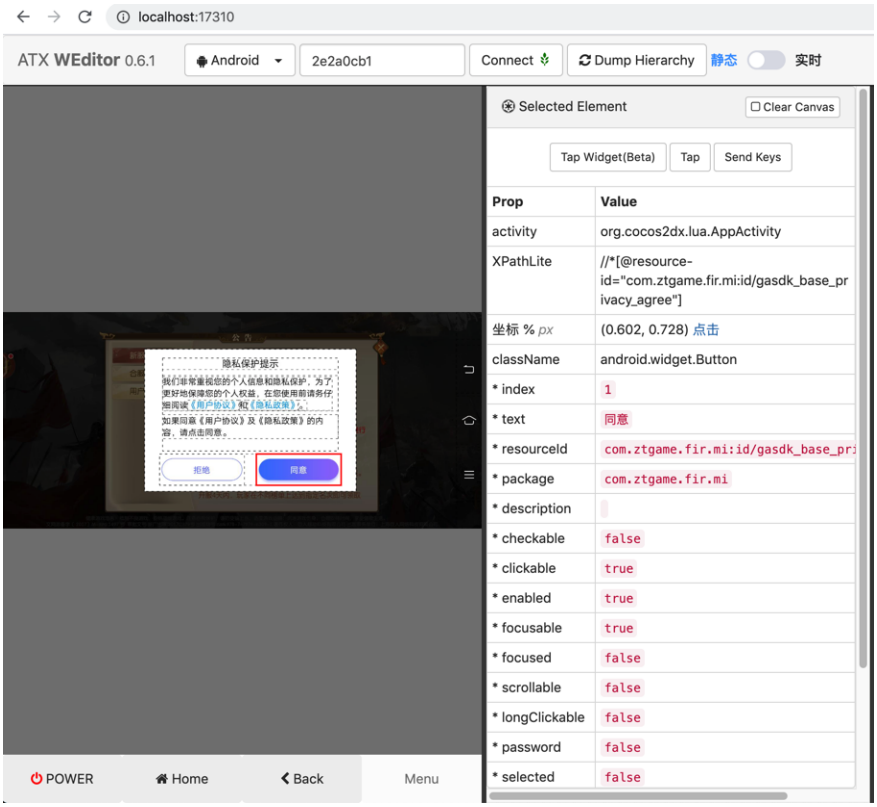


属性：

```
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.lhsl.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：

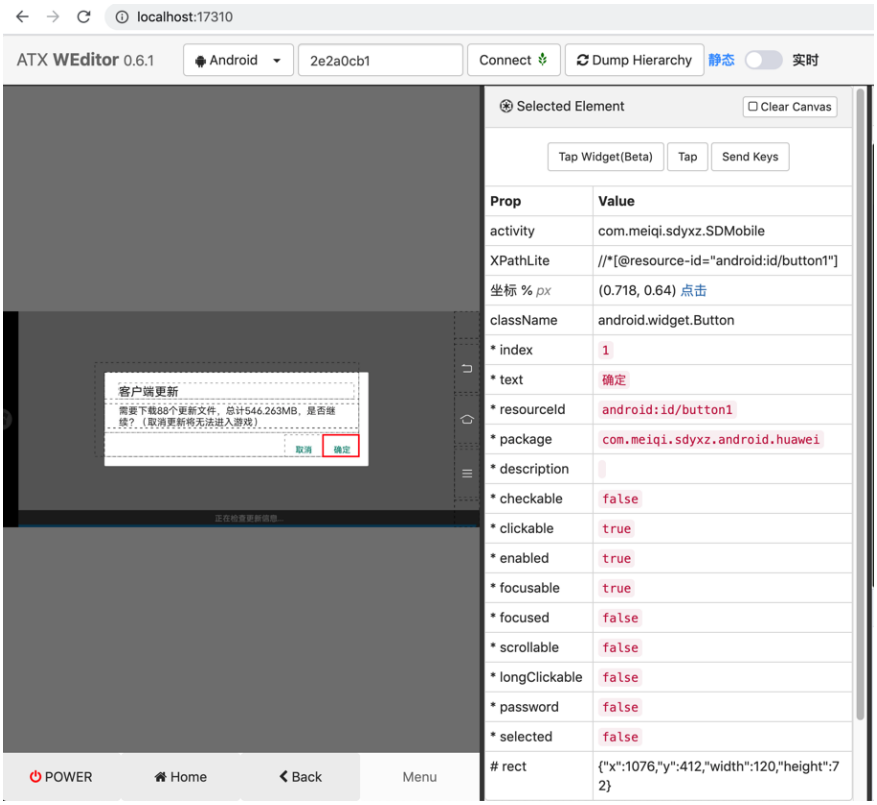


属性：

```
Prop    Value
activity    org.cocos2dx.lua.AppActivity
XPathLite    //*[@resource-id="com.ztgame.fir.mi:id/gasdk_b
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_agr
* 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_
```

客户端更新 确定

weditor截图：

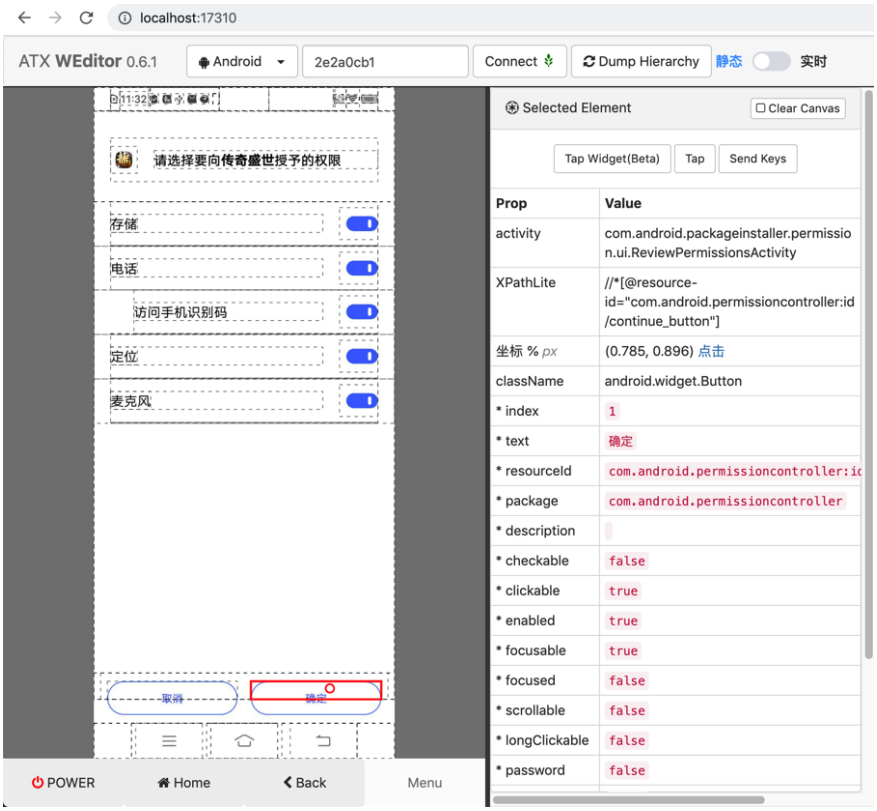


属性：

```
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：



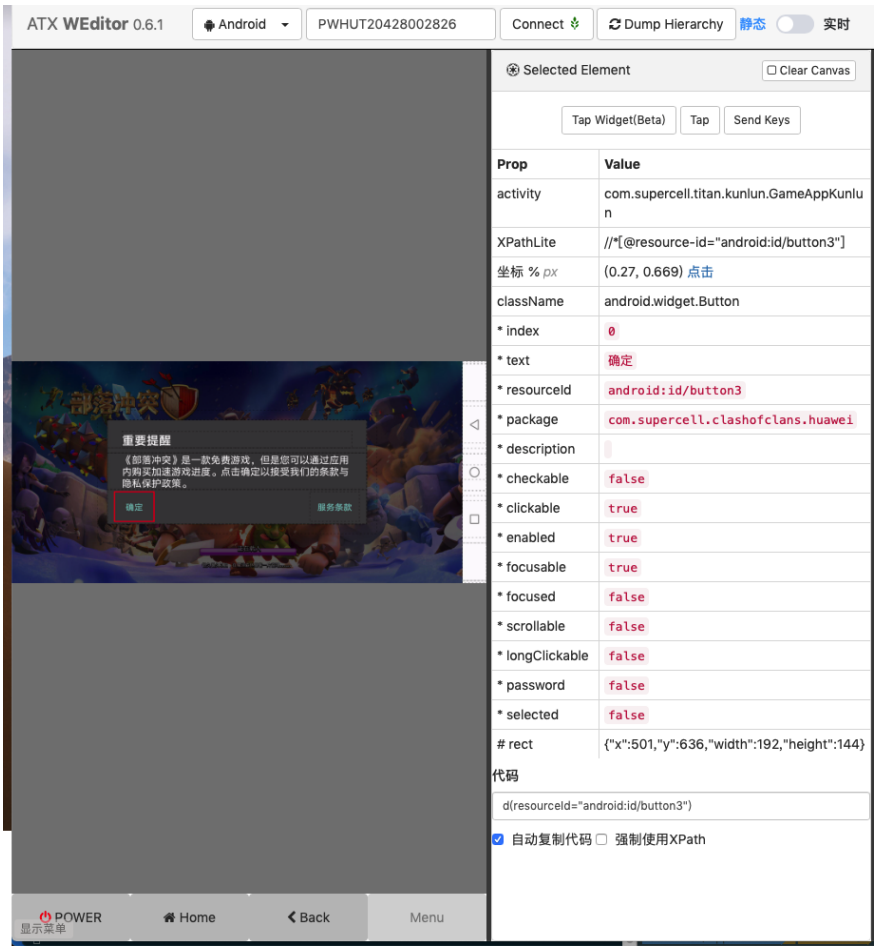
属性：

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

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}
```

（各大应用市场授权后的）广告类弹框

代码：

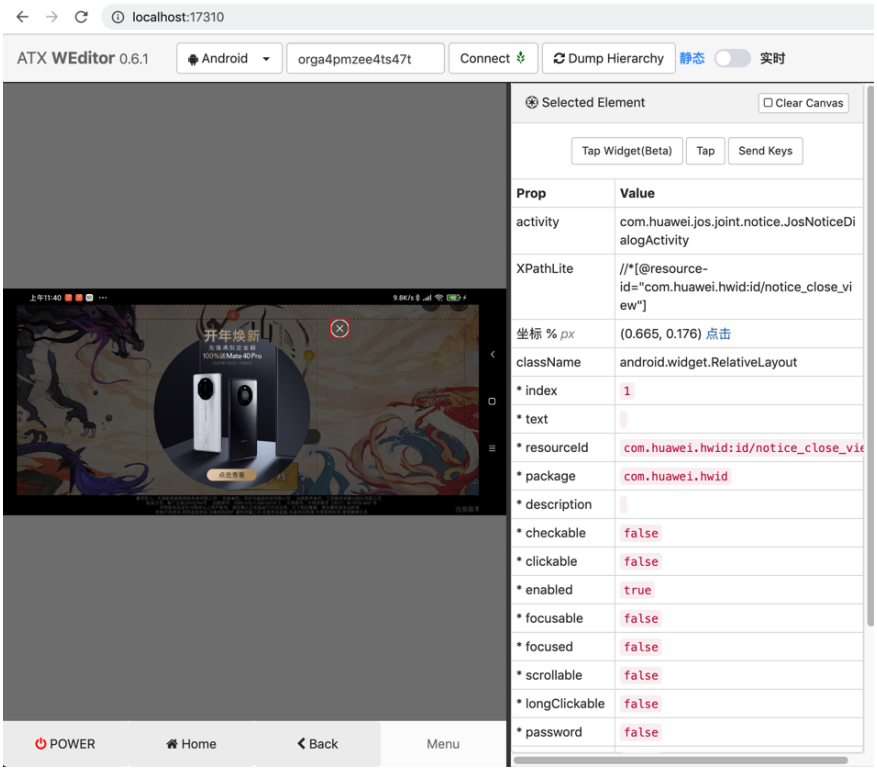
```
PopupWindow_CloseButton_Xpath_List: [
    "//android.widget.RelativeLayout[contains(@resource-id,
    "//android.widget.ImageView[contains(@resource-id, 'id/
    "//android.widget.ImageView[contains(@resource-id, 'id/
    "//android.widget.ImageView[contains(@resource-id, 'id/
    "//android.widget.ImageView[contains(@resource-id, 'id/
    "//android.widget.ImageView[contains(@resource-id, 'id/
    "//android.widget.ImageView[contains(@resource-id, 'id/
    # "//android.widget.Image[@text='7cWwAAAABJRU5ErkJggg==
    # 小米市场登录后 广告 弹框 关闭按钮
    # 1. d(text="7cWwAAAABJRU5ErkJggg==")
    # 2. d(text="2d7mODgAAAAAAAAAAAAAAAACV+wGa61esTL2CSwAA/
    # "//android.widget.Image[contains(@text, 'ggg==') and
    # 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="Iuk6V5zR2fE3Srww7HUGlCXpdxkyv
    # 改为:
    "//android.widget.Image[contains(@text, 'wAAAABJRU5Erk
    "//android.view.View[@text='知道了' and @index='0' and @
]

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.JosNoticeDialogActi
XPathLite    //*[@resource-id="com.huawei.hwid:id/notice_c

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:): ['
```

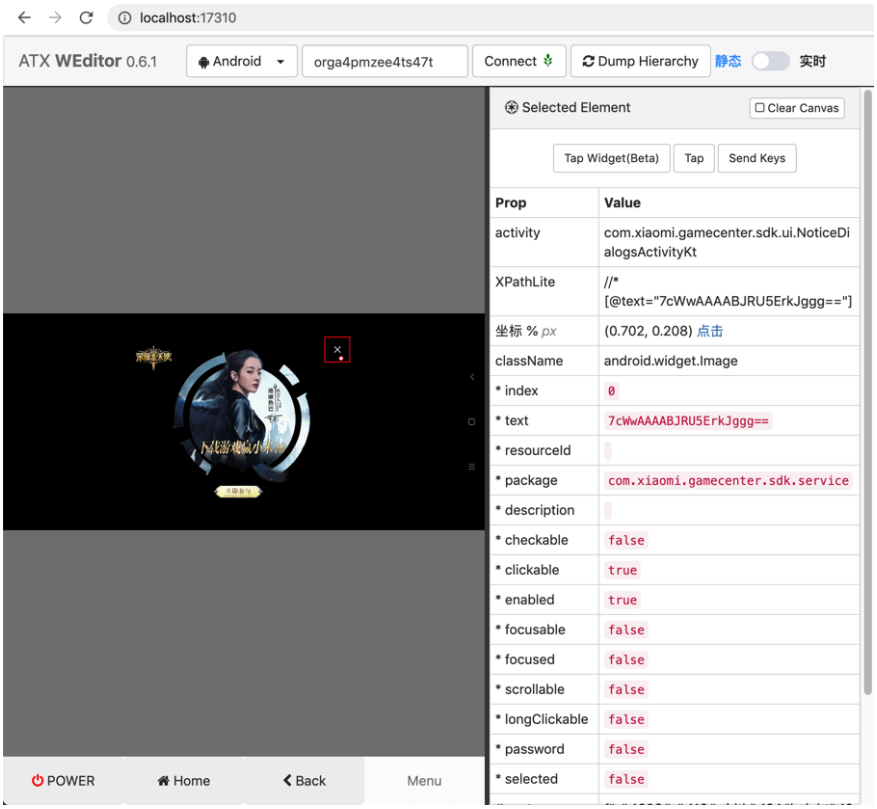
之后即可看到原先app的内容了：

小米应用市场 弹框广告

广告1

游戏app截图：

weditor截图：

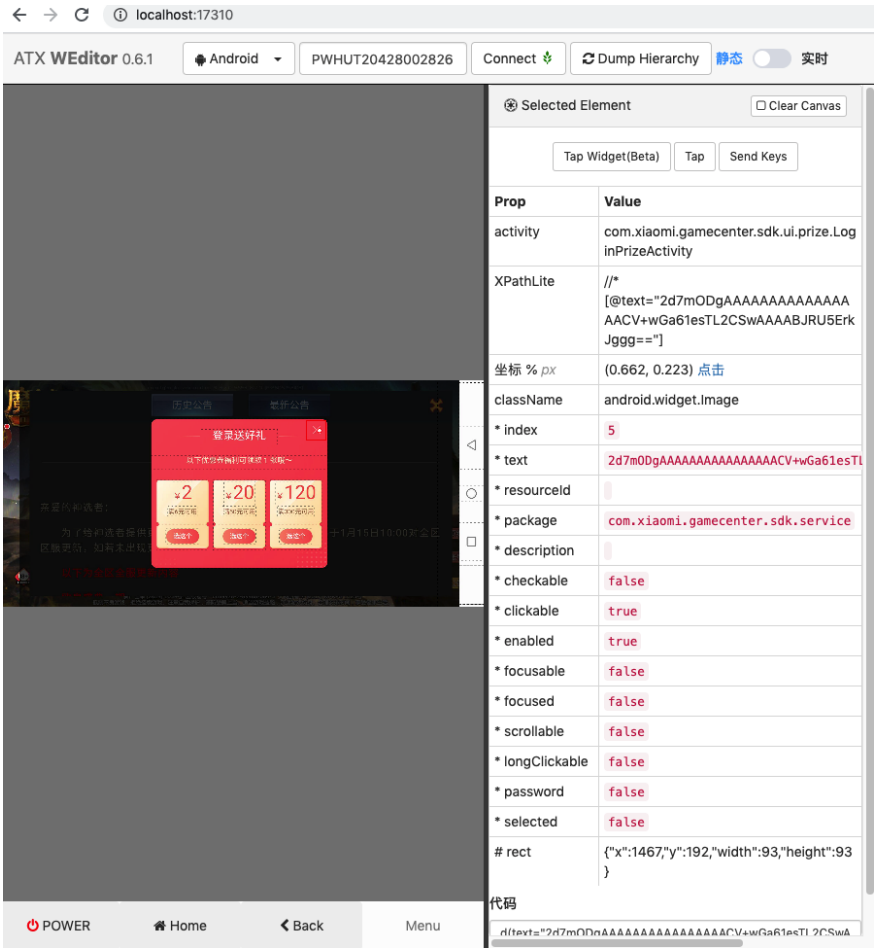


属性:

```
Prop    Value
activity    com.xiaomi.gamecenter.sdk.ui.NoticeDialogsActiv
XPathLite    //*[@text="7cWwAAAABJRU5ErkJggg=="]
坐标 % px    (0.702, 0.208) 点击
className    android.widget.Image
* index    0
* text    7cWwAAAABJRU5ErkJggg==
* 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":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.LoginPrizeAc
XPathLite    //*[@text="2d7mODgAAAAAAAAAAAAAAAACV+wGa61esTl
坐标 % px    (0.662, 0.223) 点击
className    android.widget.Image
* index    5
* text    2d7mODgAAAAAAAAAAAAAAAACV+wGa61esTL2CSwAAAABJRU5E
* 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="2d7mODgAAAAAAAAAAAAAAAACV+wGa61esTL2CSwAAAABJR
```

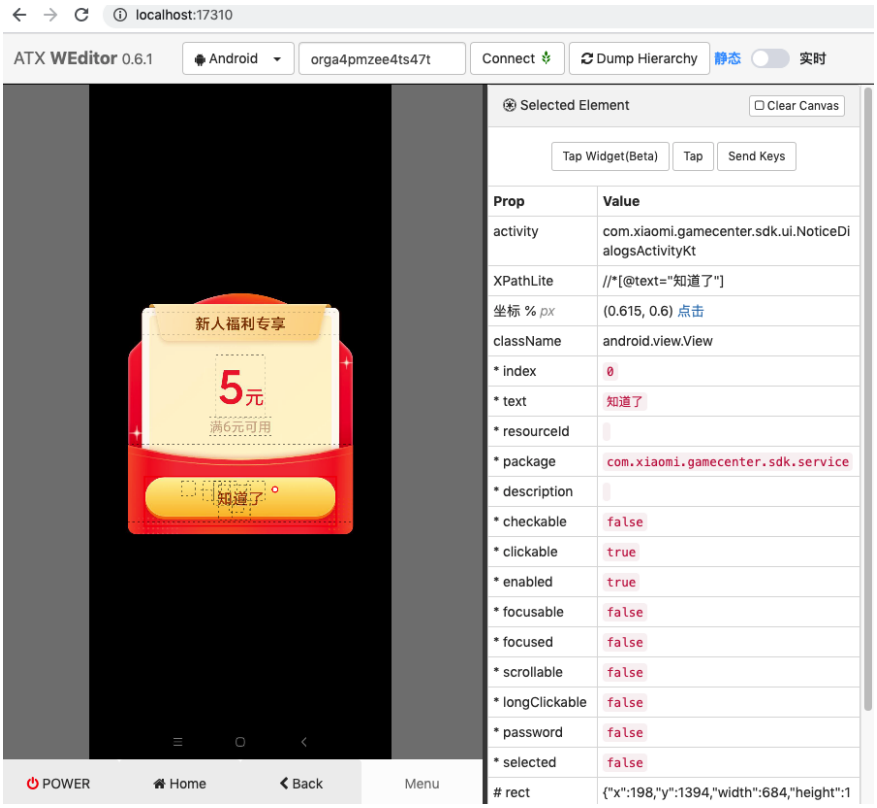
广告3

游戏： `com.yzcm.jr.mi/巨刃`

游戏app截图：


新人福利专享
5元
满6元可用
知道了

weditor截图：



属性：

```
Prop    Value
activity    com.xiaomi.gamecenter.sdk.ui.NoticeDialogsActiv
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")
```

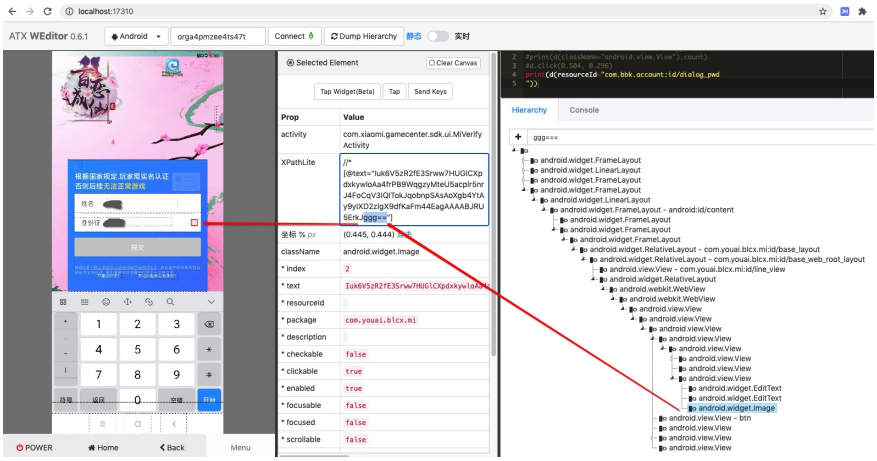
防止其他页面误判为小米广告

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="Iuk6V5zR2fE3Srww7HUGlCXpdxkywloAa4
坐标 % px    (0.445, 0.444) 点击
className    android.widget.Image
* index    2
* text    Iuk6V5zR2fE3Srww7HUGlCXpdxkywloAa4frPB9WqgzyMteU
* 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：`wAAAABJRU5ErkJggg==`

即：

```
# "//android.widget.Image[@text='7cWwAAAABJRU5ErkJggg=='
# 小米市场登录后 广告 弹框 关闭按钮
# 1. d(text="7cWwAAAABJRU5ErkJggg==")
# 2. d(text="2d7mODgAAAAAAAAAAAAAAACV+wGa61esTL2CSwAAAAE
# "//android.widget.Image[contains(@text, 'ggg==') and @
# 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="Iuk6V5zR2fE3Srww7HUGlCXpdxkywl
# 改为:
"//android.widget.Image[contains(@text, 'wAAAABJRU5ErkJg
```

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：

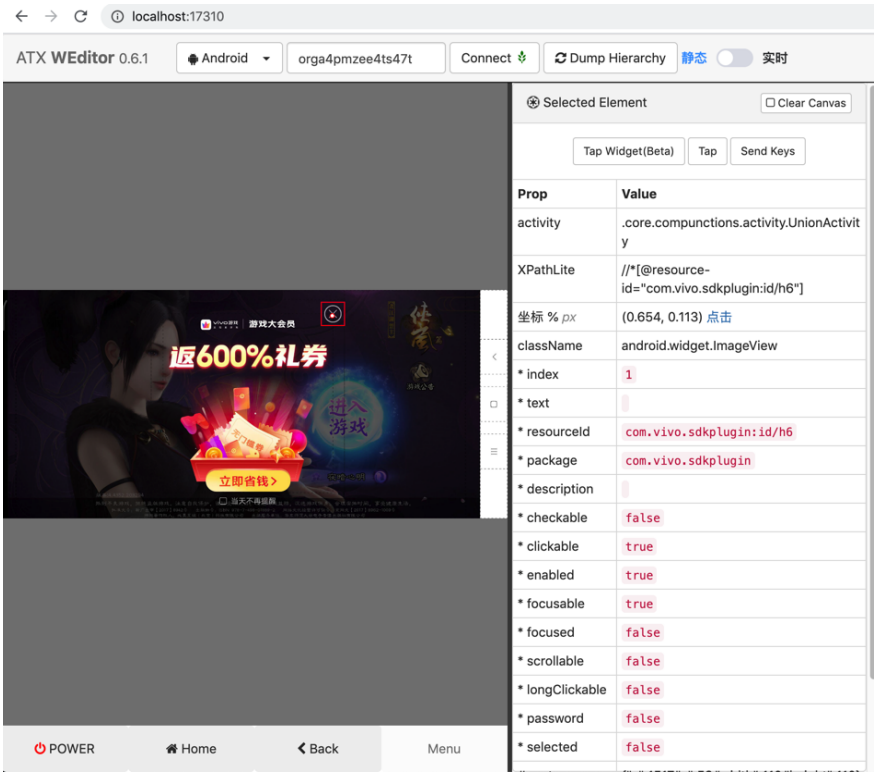
【已解决】用uiautomator2自动点击关闭小米应用市场登录后的广告弹框

Vivo应用市场 弹框广告

广告1

游戏app截图：

weditor截图：



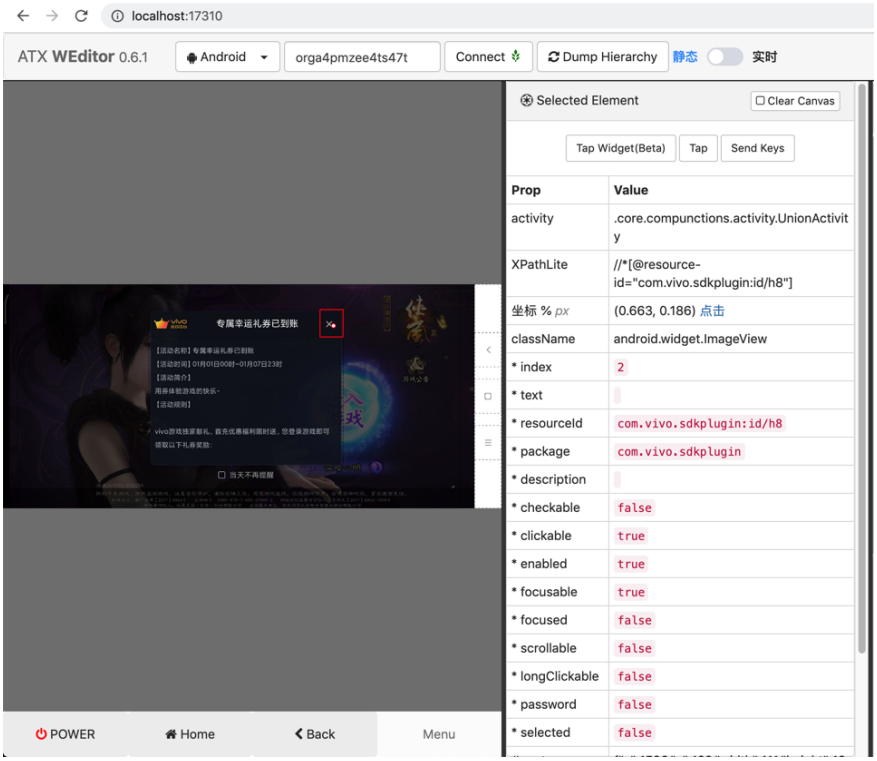
属性：

```
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：



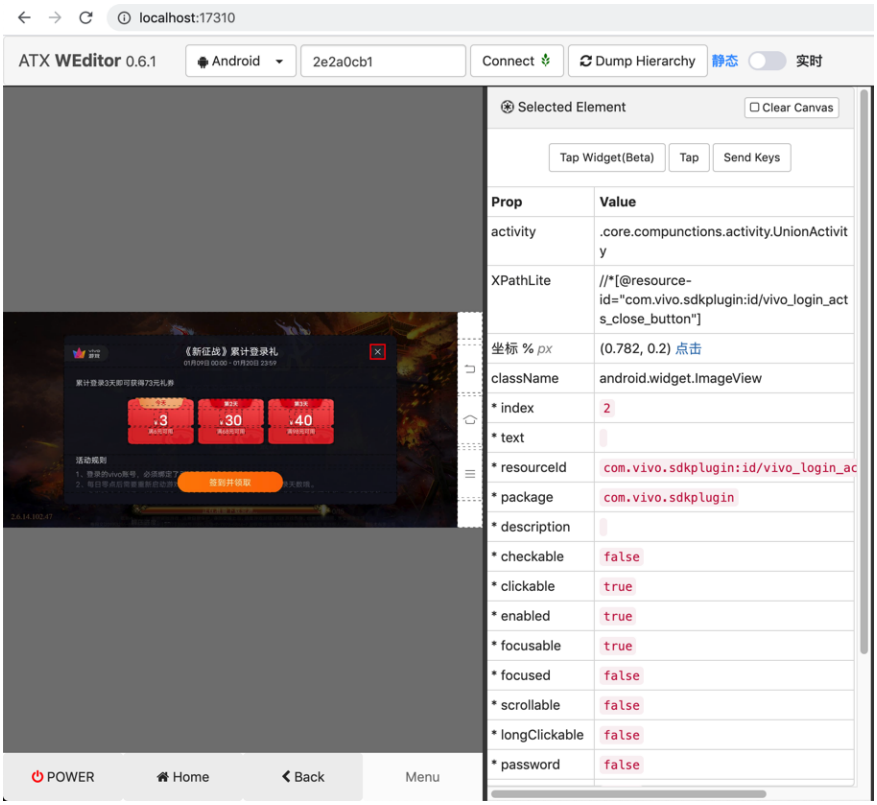
属性：

```
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：

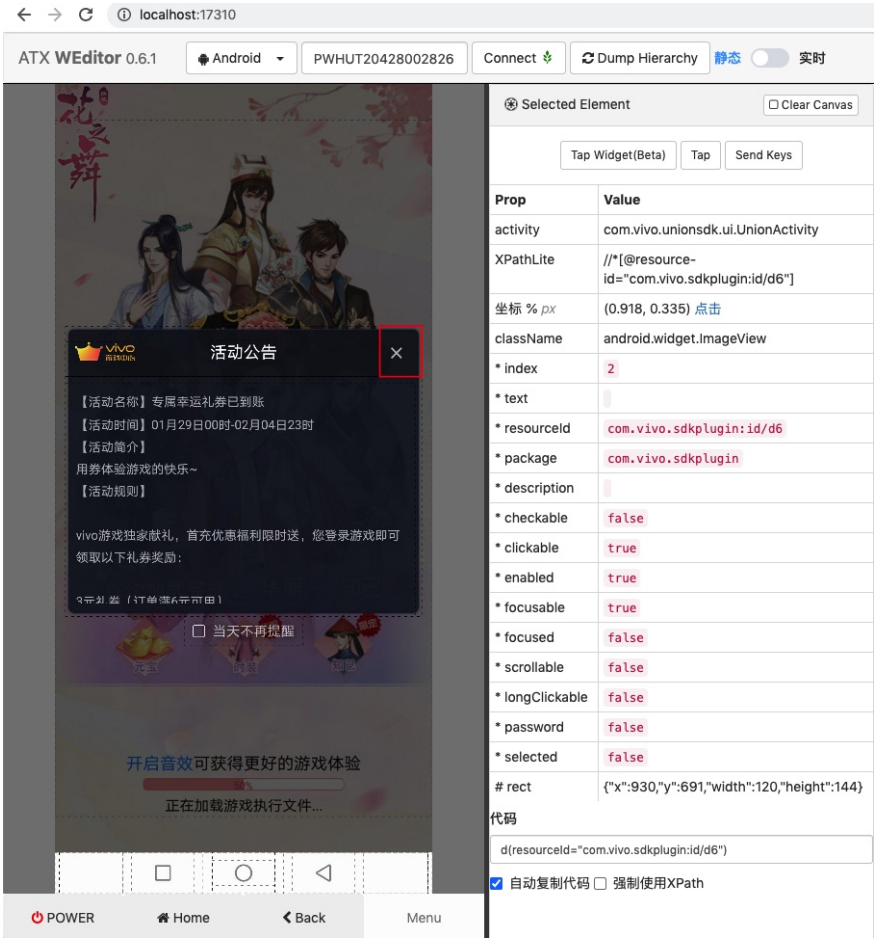
```
Prop    Value
activity    .core.compunctions.activity.UnionActivity
XPathLite    //*[@resource-id="com.vivo.sdkplugin:id/vivo_
坐标 % px    (0.782, 0.2) 点击
className    android.widget.ImageView
* index    2
* text
* resourceId    com.vivo.sdkplugin:id/vivo_login_acts_close
* package    com.vivo.sdkplugin
* 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_acts_cl
```

广告4

游戏app截图：


花之舞
vivo 游戏中心
活动公告
【活动名称】专属幸运礼券已到账
【活动时间】01月29日00时-02月04日23时
【活动简介】
用券体验游戏的快乐~
【活动规则】
vivo游戏独家献礼，首充优惠福利限时送，您登录游戏即可领取以下礼券奖励：
当天不再提醒
元宝
时装
限定
知己
开启音效可获得更好的游戏体验
50%
正在加载游戏执行文件...

weditor截图：



核心属性：

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

Vivo手机中安装应用时，自动输入账号密码

背景：

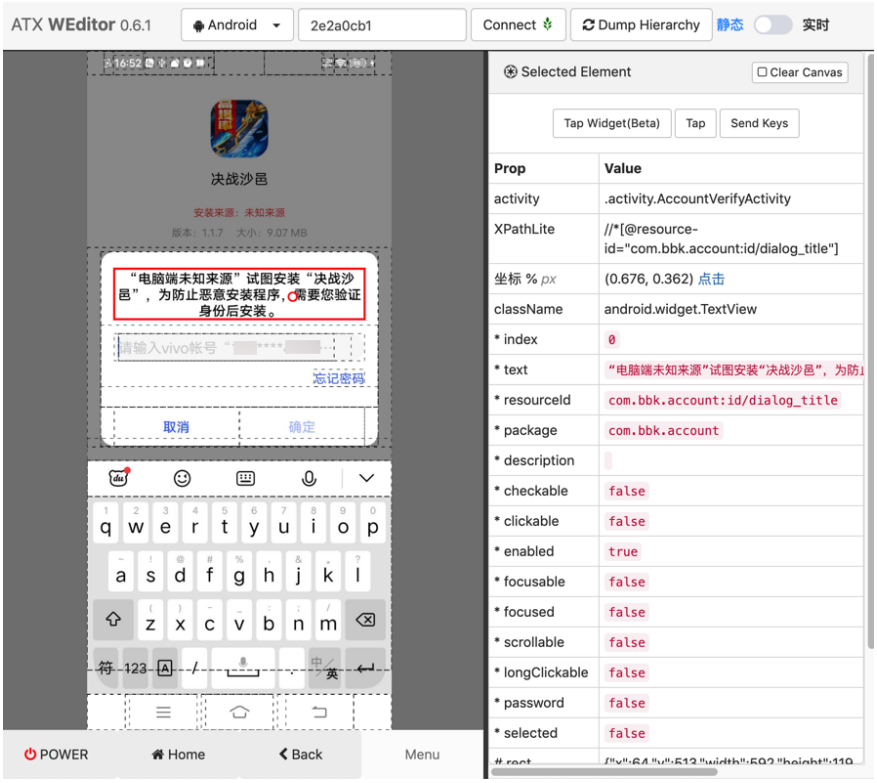
用 `adb` 去给安卓手机安卓 `apk` ，比如：

`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_t
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
* 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")
```

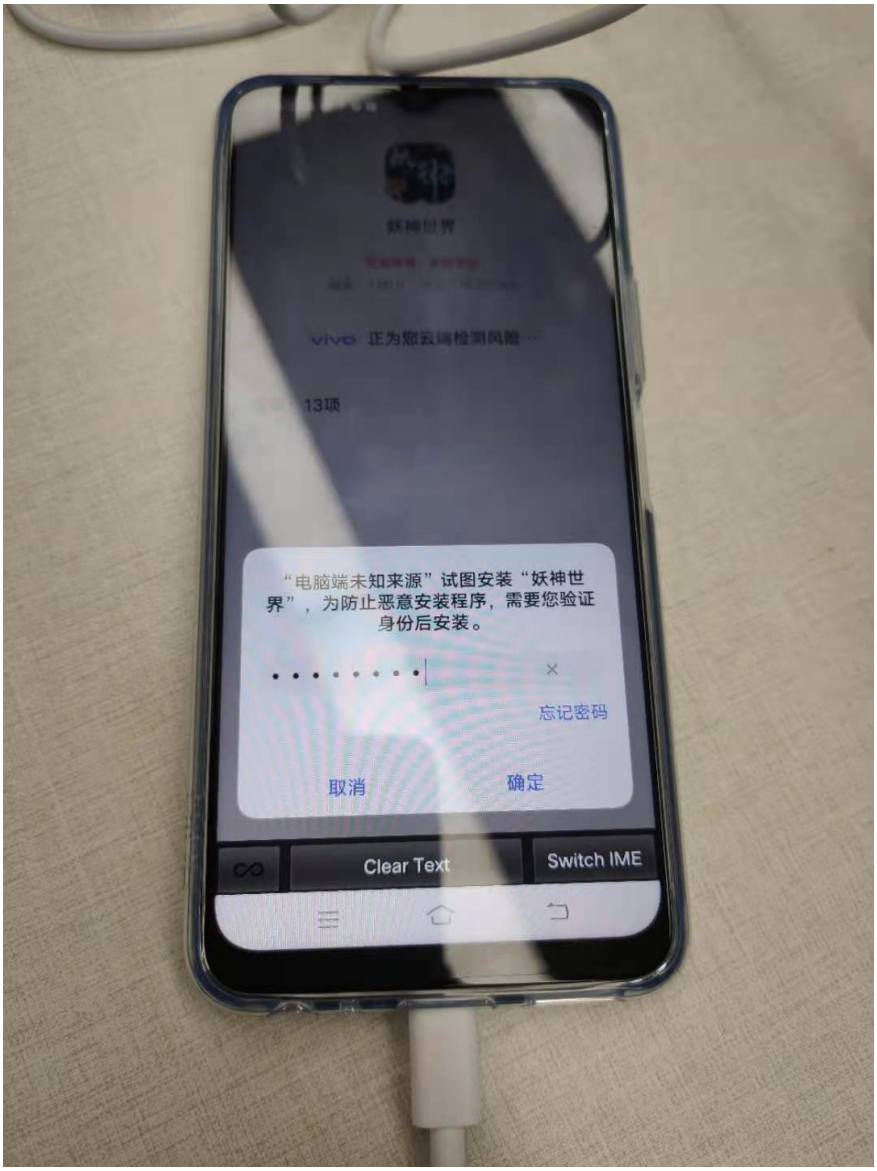
经过调试，最终实现了 自动化输入账号和密码，得以自动继续安装。

代码：

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

Vivo_Password_Input_Xpath: "//android.widget.LinearLayout[@

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

def selectorSetText(self, curXpathSelector, inputText):
    # Special: add click to try workaround for 360 pwd Edit
    # curXpathSelector.click()
    # curXpathSelector.clear_text()
    selectorSetTextResp = curXpathSelector.set_text(inputTe
    logging.debug("selectorSetTextResp=%s", selectorSetText
    # 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
    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_In
    logging.debug("pwdDiaglogSelector=%s", pwdDiaglogSelect
    # PwdDiaglogSelector=XPathSelector(//android.widget.Lin
    logging.info("%s to found password dialog", pwdDiaglogS
    # selectorSetTextResp = pwdDiaglogSelector.set_text(Viv
    # logging.debug("selectorSetTextResp=%s", selectorSetTe
    # # 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
    # 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, classN
    logging.debug("okButtonElement=%s", okButtonElement)
    logging.info("%s to found %s button", okButtonElement.e
    if okButtonElement.exists:
        okButtonElement.click()
        logging.info("Clicked 确定 for vivo password")

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

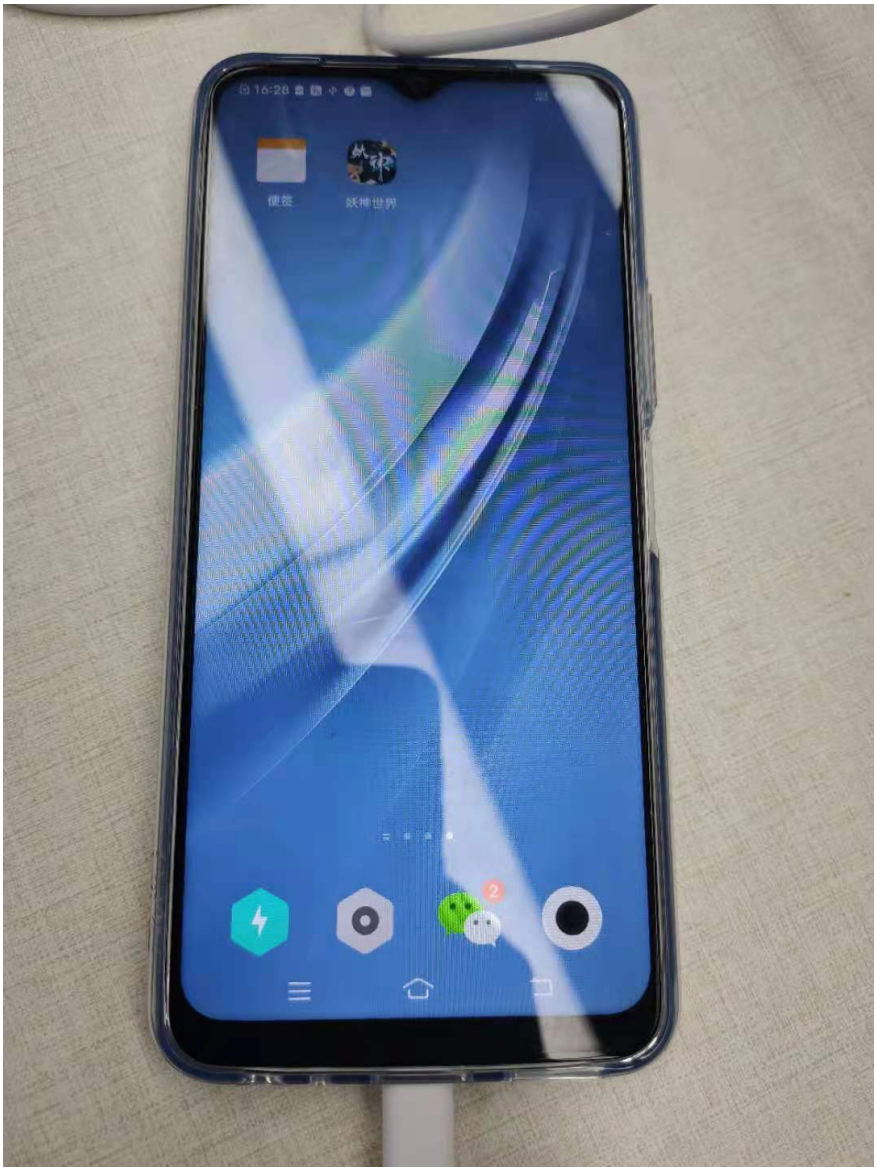
对应自动化操作期间的手机截图：

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.v
[201216 09:34:37][DevicesMethods.py 245] Try auto input viv
[201216 09:34:37][DevicesMethods.py 250] True to found pass
[201216 09:34:39][DevicesMethods.py 254] Has input passwor
[201216 09:34:39][DevicesMethods.py 263] True to found 确定
[201216 09:34:40][DevicesMethods.py 266] Clicked 确定 for v
[I 201216 09:34:46 watcher:255] XPath(hook:继续安装): ['继续
Success
```

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

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。

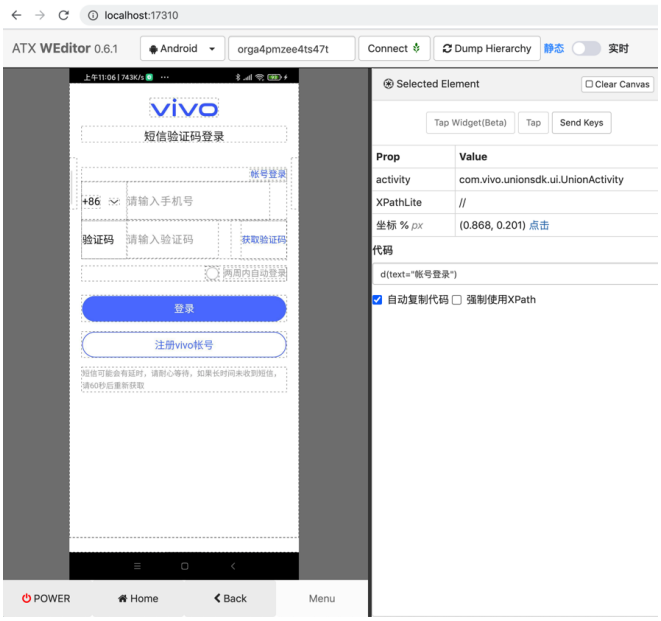
所以写代码将此过程自动化。

具体过程：

- 短信验证码登录页


上午10:54 | 1.3K/s
vivo
短信验证码登录
帐号登录
+86
请输入手机号
验证码
请输入验证码
获取验证码
两周内自动登录
登录
注册vivo帐号
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

- 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

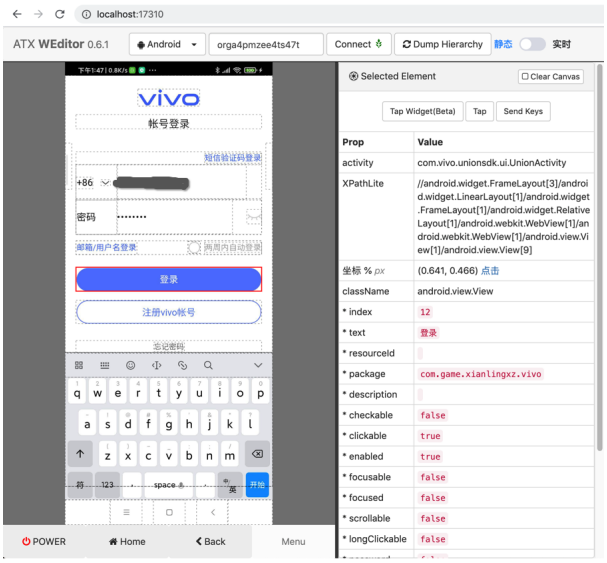
-
- 支持已经输入手机号，再次重新输入手机号
- 此处输入手机号会触发输入法切换到 fastIME

- 请输入密码：输入密码


下午1:43 | 81.8K/s
vivo
帐号登录
短信验证码登录
+86
密码
邮箱/用户名登录
两周内自动登录
登录
注册vivo帐号
忘记密码
符
123
space
中/英
开始

-
  - 真机效果

- 
上午10:57 | 0.2K/s
vivo
帐号登录
短信验证码登录
+86
密码
邮箱/用户名登录
两周内自动登录
登录
注册vivo帐号
小米安全键盘
1 2 3 4 5 6 7 8 9 0
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
!?#
123
空格
@
- 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(inputTe
    logging.info("selectorSetTextResp=%s", selectorSetTextR
    # 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
    u2Dev.set_fastinput_ime(False)

##########################################################
# for Redmi 10X: auto do vivo account login, input phone an
##########################################################

def autoDoVivoLogin(u2Dev):
    doScreenshot(u2Dev)

    RegVivoAccountStr = "注册vivo帐号"
    regVivoAccountXpath = """//android.view.View[@text="%s"
    # '//android.view.View[@text="注册vivo帐号"]'
    regVivoAccountSelector = u2Dev.xpath(regVivoAccountXpat
    if regVivoAccountSelector.exists:
        logging.info("Found %s", RegVivoAccountStr)
        accountLoginStr = "帐号登录"
        # accountLoginXpath = """//android.view.View[@text=
        # #
        # accountLoginSelector = u2Dev.xpath(accountLoginXp
        # if accountLoginSelector.exists:
        #     logging.info("Found %s", accountLoginStr)
        #     accountLoginSelector.click()
        # accountLoginElement = u2Dev(text=accountLoginStr,
        accountLoginElement = u2Dev(text=accountLoginStr, c
        logging.info("accountLoginElement=%s", accountLogin
        logging.info("accountLoginElement.exists=%s", accou
        if accountLoginElement.exists:
            accountLoginElement.click()
            logging.info("Has clicked %s button", accountLo

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

        # phoneXpath = """//android.widget.EditText[@text="
        phoneXpath = """//android.widget.EditText[@index="1
        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_Acco
```

```
            logging.info("Has input vivo account phone numb
        else:
            logging.warning("Not found 请输入手机号")

        passwordStr = "请输入密码"
        # passwordXpath = """//android.widget.EditText[@tex
        passwordXpath = """//android.widget.EditText[@index
        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_P
            logging.info("Has input vivo password")
        else:
            logging.warning("Not found %s", passwordStr)

        loginStr = "登录"
        # loginXpath = """//android.view.View[@text="登录"
        loginXpath = """//android.view.View[@text="登录" an
        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.D
。。。
    autoDoVivoLogin(u2Dev)
```

合并到项目后：

```
Vivo_Register_Vivo_Account_Xpath: "//android.view.View[@tex

self.driver.watcher.when(self.config["Vivo_Register_Vivo_Ac

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"
    # #
    # accountLoginSelector = self.driver.xpath(accountLogin
    # if accountLoginSelector.exists:
    #     logging.info("Found %s", accountLoginStr)
    #     accountLoginSelector.click()
    # accountLoginElement = self.driver(text=accountLoginSt
    accountLoginElement = self.driver(text=accountLoginStr,
    logging.debug("accountLoginElement=%s", accountLoginEle
    logging.debug("accountLoginElement.exists=%s", accountL
    if accountLoginElement.exists:
        accountLoginElement.click()
        logging.info("Has clicked %s button", accountLoginS

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

    # phoneXpath = """//android.widget.EditText[@text="" an
    phoneXpath = """//android.widget.EditText[@index="5" an
    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["Vi
        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="%
    passwordXpath = """//android.widget.EditText[@index="2"
    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
        logging.info("Has input vivo password")
    else:
```

```
            logging.warning("Not found %s", passwordStr)

        loginStr = "登录"
        # loginXpath = """//android.view.View[@text="登录" and
        loginXpath = """//android.view.View[@text="登录" and @i
        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


下午2:24 | 0.0K/s
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


下午2:25 | 0.0K/s
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
号 点 个 的 元 后
1 2 3
4 5 6
7 8 9
符号 返回 0 空格 开始

- 验证后，会自动返回

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。

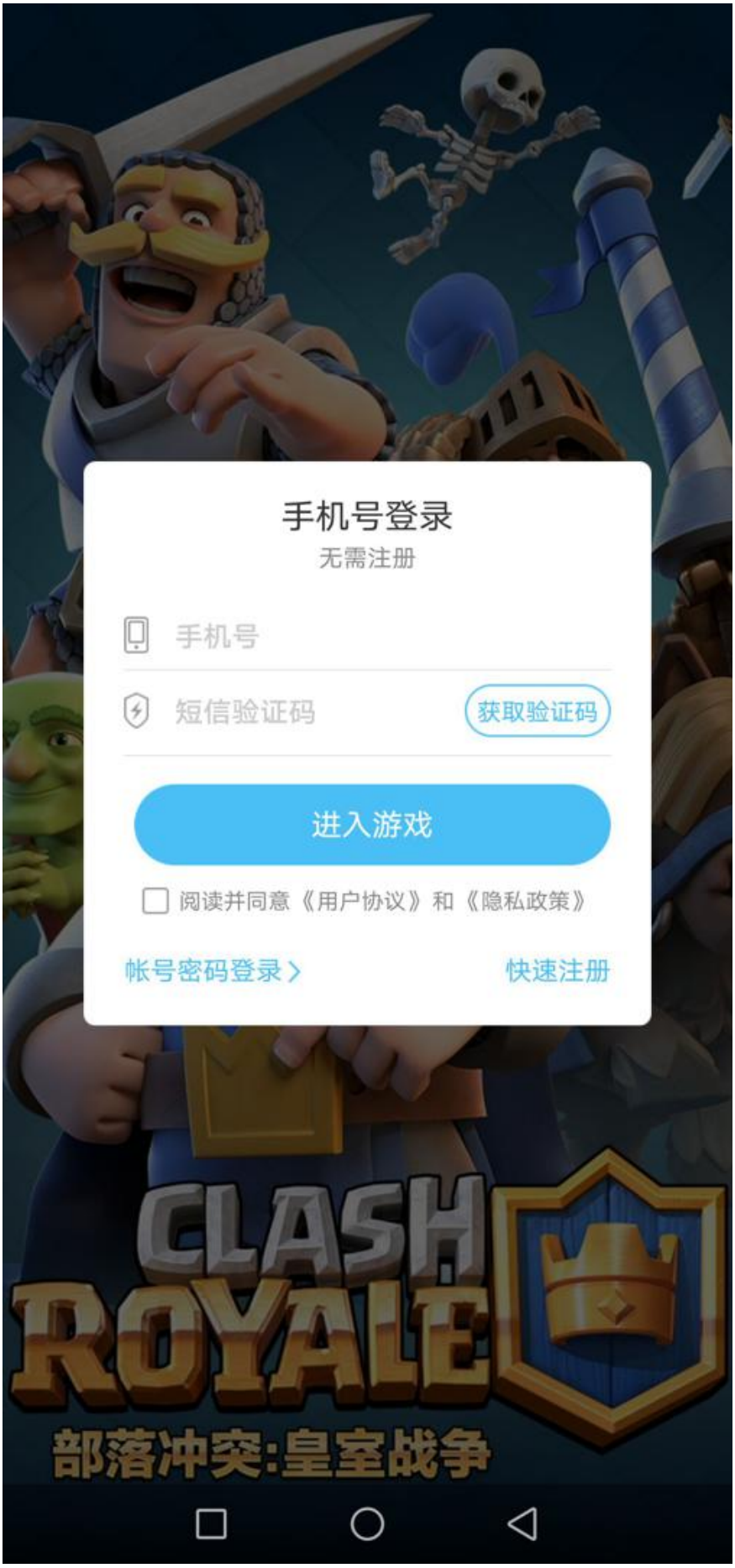
其中在注册和登录期间，会涉及到：

360账号的注册和登录

比如：

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

弹框：


手机号登录
无需注册
手机号
短信验证码
获取验证码
进入游戏
阅读并同意《用户协议》和《隐私政策》
帐号密码登录 >
快速注册
CLASH ROYALE
部落冲突:皇室战争

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.Text
# Qihoo360_Login_ReadAndAgree_Xpath: "//android.widget.Text
Qihoo360_PasswordLogin_Xpath: "//android.widget.TextView[@

self.driver.watcher.when(Qihoo360_PasswordLogin_Xpath).cal

def selectorSetText(self, curXpathSelector, inputText):
    # Special: add click to try workaround for 360 pwd Edi
    # curXpathSelector.click()
    # curXpathSelector.clear_text()
    selectorSetTextResp = curXpathSelector.set_text(inputTe
    logging.debug("selectorSetTextResp=%s", selectorSetText
    # 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
    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
    # reg360AccountPwdXpath = """//android.widget.TextView
    reg360AccountPwdXpath = """//android.widget.TextView[@
    # '//android.widget.TextView[@resource-id="com.qihoo.ga
    reg360AccountPwdSelector = self.driver.xpath(reg360Acc
    if reg360AccountPwdSelector.exists:
        # doScreenshot(u2Dev)
        # logging.info("Found %s", AccountPwdStr)
        reg360AccountPwdSelector.click()
        time.sleep(0.1)
        logging.info("Has clicked %s button", AccountPwdSt
    else:
        logging.warning("Not found %s button", AccountPwdSt

    SwitchLoginTypeStr = "切换登录方式"
    switchLoginTypeXpath = """//android.widget.TextView[@re
    switchLoginTypeSelector = self.driver.xpath(switchLogin
    isInAccountPwdLoginPage = switchLoginTypeSelector.exist
    if isInAccountPwdLoginPage:
        logging.info("Found %s", SwitchLoginTypeStr)

    if not isInAccountPwdLoginPage:
        ShortSmsVerifyCodeLoginStr = "短信验证码登录 "
        # shortSmsVerifyCodeLoginXpath = """//android.widge
        shortSmsVerifyCodeLoginXpath = """//android.widget.
```

```
        shortSmsVerifyCodeLoginSelector = self.driver.xpath
        isInAccountPwdLoginPage = shortSmsVerifyCodeLoginSe
        if isInAccountPwdLoginPage:
            logging.info("Found %s", ShortSmsVerifyCodeLogi

    if not isInAccountPwdLoginPage:
        AccountLogin360Str = "360帐号登录"
        accountLogin360Xpath = """//android.widget.TextVie
        accountLogin360Selector = self.driver.xpath(accoun
        isInAccountPwdLoginPage = accountLogin360Selector.e
        if isInAccountPwdLoginPage:
            logging.info("Found %s", AccountLogin360Str)

    if not isInAccountPwdLoginPage:
        logging.error("Not in 360 account and password log
        return

    logging.info("In 360 account and password login page, c

    AccountStr = "360帐号/手机号/邮箱"
    # accountXpath = """//android.widget.EditText[@resource
    accountXpath = """//android.widget.EditText[@resource-
    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_
            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
    # TODO: try xiaomi 10X is or or not
    return

    PasswordStr = "密码"

    # # passwordXpath = """//android.widget.EditText[@resou
    # # passwordXpath = """//android.widget.EditText[@resou
    # # passwordXpath = """//android.widget.EditText[@resou
    # passwordXpath = """//android.widget.EditText[@resourc
    # passwordSelector = self.driver.xpath(passwordXpath)
    # if passwordSelector.exists:
    #     logging.info("Found %s", PasswordStr)
    #     # curPassword = passwordSelector.text
    #     # if curPassword and (curPassword == Qihoo360_Pas
```

```
    #     #     logging.info("Already inputed 360 password'
    #     # else:
    #     self.selectorSetText(passwordSelector, Qihoo360_P
    #     logging.info("Has input 360 password")
    # else:
    #     logging.warning("Not found %s", PasswordStr)

    passwordElement = self.driver(className="android.widget
    logging.info("passwordElement.exists=%s", passwordEleme
    if passwordElement.exists:
        passwordElement.set_text(Qihoo360_Password)
        logging.info("Has input 360 password")
    else:
        logging.warning("Not found %s", PasswordStr)

    AggreeStr = "阅读并同意用户协议"
    # aggreeCheckboxElement = u2Dev(className="android.widg
    # if aggreeCheckboxElement.exists:
    #     aggreeCheckboxElement.click()
    aggreeCheckboxXpath = """//android.widget.CheckBox[@res
    aggreeCheckboxSelector = self.driver.xpath(aggreeCheckb
    if aggreeCheckboxSelector.exists:
        aggreeCheckboxSelector.click()
        time.sleep(0.1)
        logging.info("Has clicked %s checkbox", AggreeStr)
    else:
        logging.warning("Not found %s checkbox", AggreeStr)
        return

    IntoGameStr = "进入游戏"
    IntoGameXpath = """//android.widget.TextView[@resource-
    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
    PleaseAggreePopupStr = "请同意用户协议和隐私政策"
    PleaseAggreePopupXpath = """//android.widget.TextView[@
    PleaseAggreePopupSelector = self.driver.xpath(PleaseAgg
    if PleaseAggreePopupSelector.exists:
        logging.info("Found %s popup", PleaseAggreePopupStr

        PositiveButtonStr = "确定"
        positiveButtonXpath = """//android.widget.TextView
        positiveButtonSelector = self.driver.xpath(positive
        if positiveButtonSelector.exists:
            positiveButtonSelector.click()
```

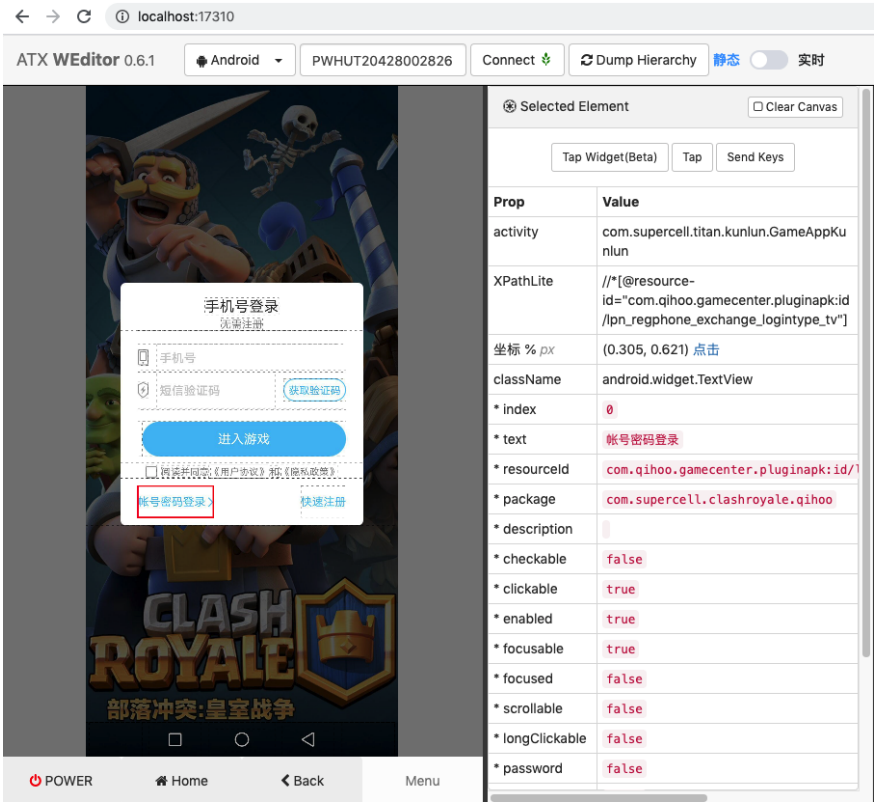
```
            logging.info("Has clicked %s button", PositiveB
            time.sleep(0.1)

            # do second time
            IntoGameStr = "进入游戏"
            IntoGameXpath = """//android.widget.TextView[@t
            loginSelector = self.driver.xpath(IntoGameXpath
            if loginSelector.exists:
                loginSelector.click()
                time.sleep(0.1)
                logging.info("Has clicked %s button", IntoG
            else:
                logging.warning("Not found %s button", Into
    else:
        logging.info("Not found %s", PleaseAggreePopupStr)

    # wait doing login
    LoginingStr = "正在登录..."
    loginingXpath = """//android.widget.TextView[@text="%s"
    loginingSelector = self.driver.xpath(loginingXpath)
    isLogining = loginingSelector.exists
    while isLogining:
        loginingSelector = self.driver.xpath(loginingXpath)
        isLogining = loginingSelector.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[@re
    verifyCodeErrorSelector = self.driver.xpath(verifyCodeE
    if verifyCodeErrorSelector.exists:
        logging.info("Found %s notice", VerifyCodeErrorStr)

        # locate input cursor
        PleaseInputRightVerifyCodeStr = "请输入右侧的验证码"
        # pleaseInputRightVerifyCodeXpath = """//android.wi
        pleaseInputRightVerifyCodeXpath = """//android.widg
        pleaseInputRightVerifyCodeSelector = self.driver.xp
        if pleaseInputRightVerifyCodeSelector.exists:
            pleaseInputRightVerifyCodeSelector.click()
            logging.info("Has clicked %s", PleaseInputRight
        else:
            logging.warning("Not found %s button", PleaseIn
    else:
        logging.warning("Not found %s button", VerifyCodeEr
```

相关截图

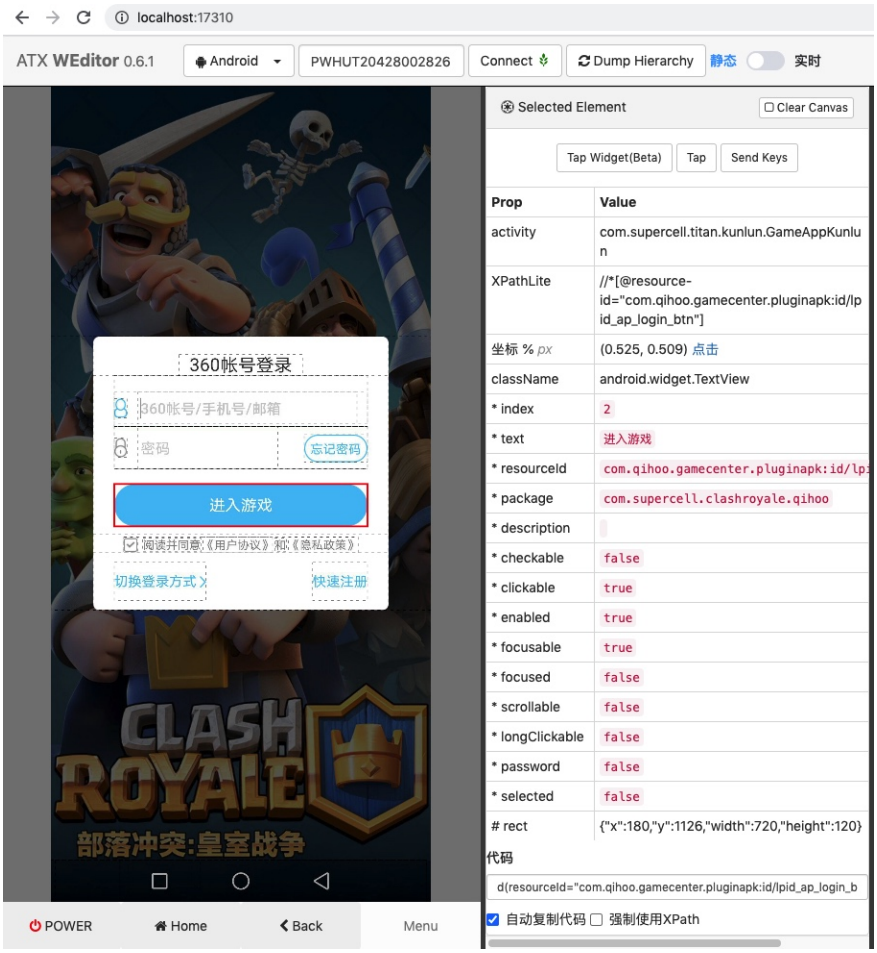
调试时

weditor截图：

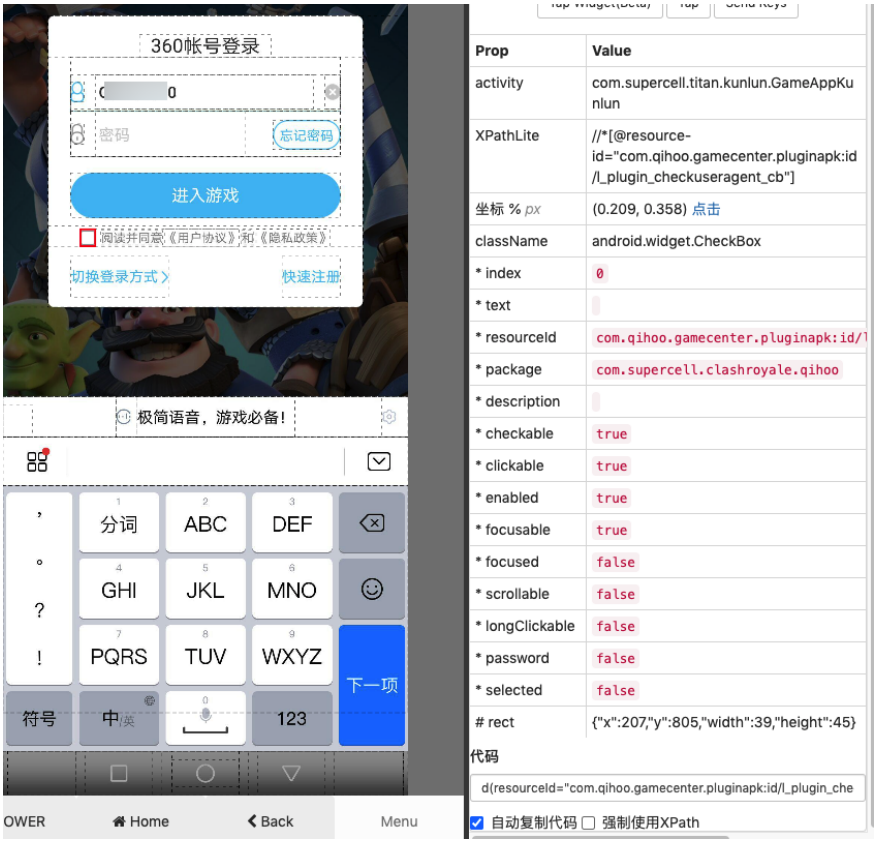
点击 账号和密码登录：



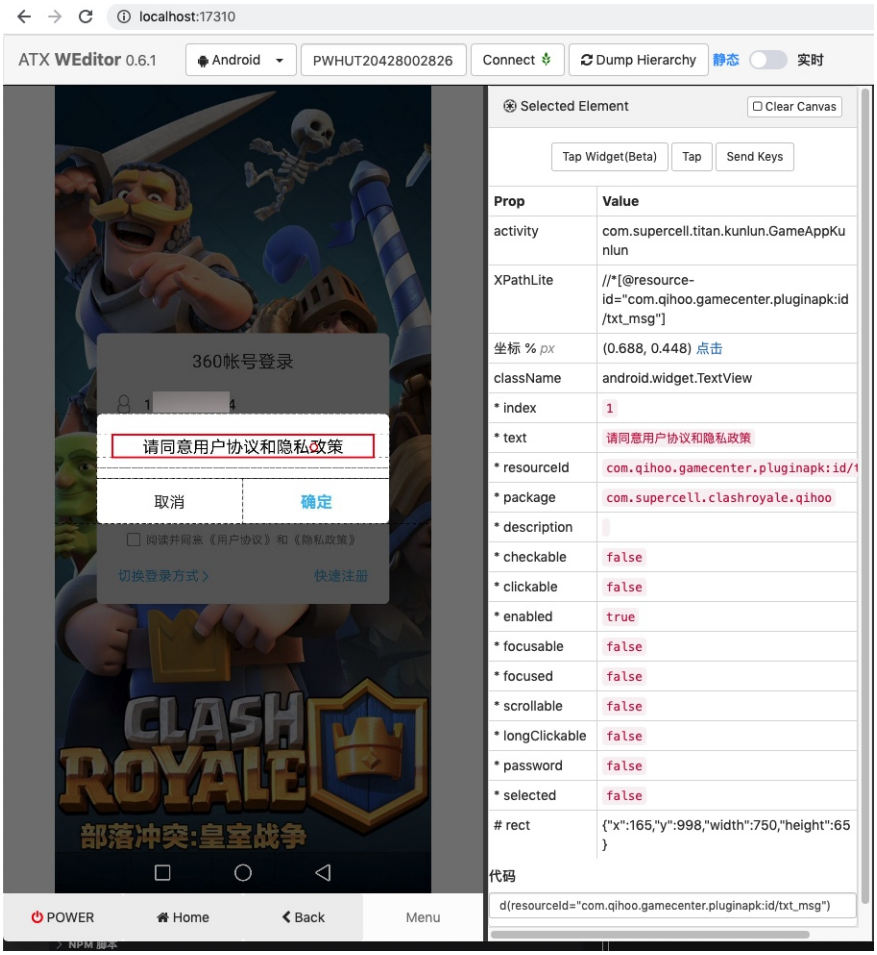
切换到 360账号登录：


localhost:17310
ATX WEditor 0.6.1
Android
PWHUT20428002826
Connect
Dump Hierarchy
静态
实时
360帐号登录
360帐号/手机号/邮箱
密码
忘记密码
进入游戏
阅读并同意《用户协议》和《隐私政策》
切换登录方式
快速注册
CLASH ROYALE
部落冲突:皇室战争
POWER
Home
Back
Menu
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/lpid_ap_login_btn"]
坐标 % px
(0.525, 0.509) 点击
className
android.widget.TextView
* index
2
* text
进入游戏
* resourceId
com.qihoo.gamecenter.pluginapk:id/lp
* package
com.supercell.clashroyale.qihoo
* description
* checkable
false
* clickable
true
* enabled
true
* focusable
true
* focused
false
* scrollable
false
* longClickable
false
* password
false
* selected
false
rect
{"x":180,"y":1126,"width":720,"height":120}
代码
d(resourceId="com.qihoo.gamecenter.pluginapk:id/lpid_ap_login_b
自动复制代码
强制使用XPath

已输入360账号:

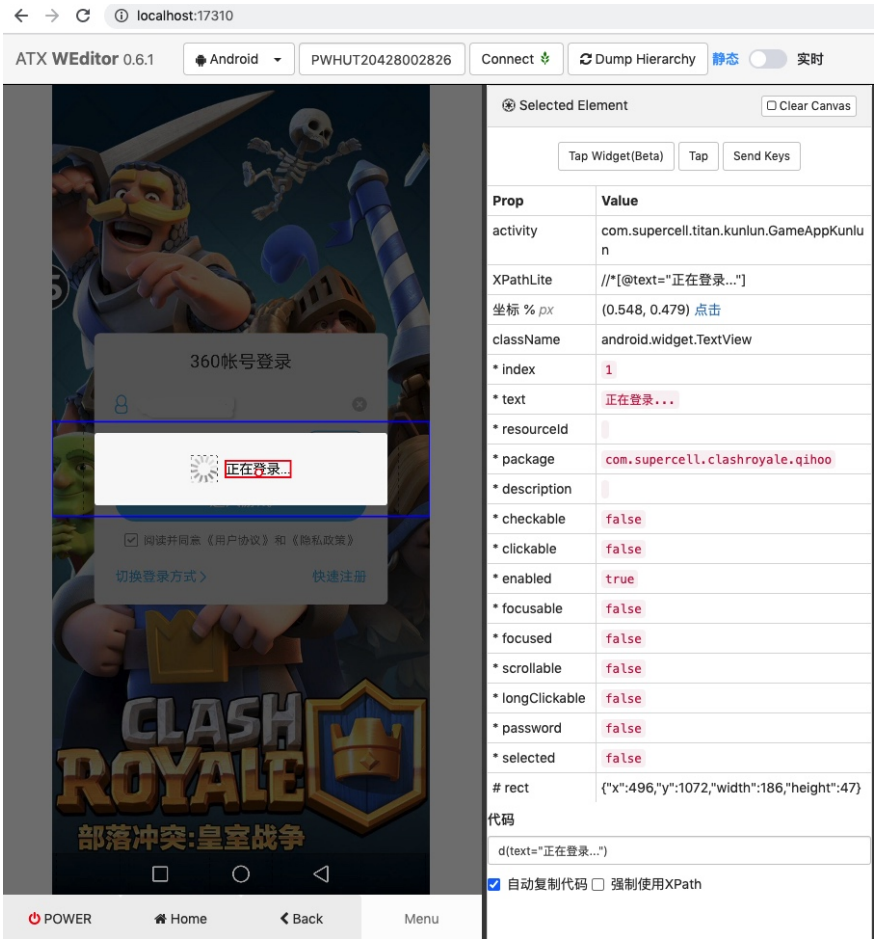


点击登录时，提示：请同意用户协议和隐私政策

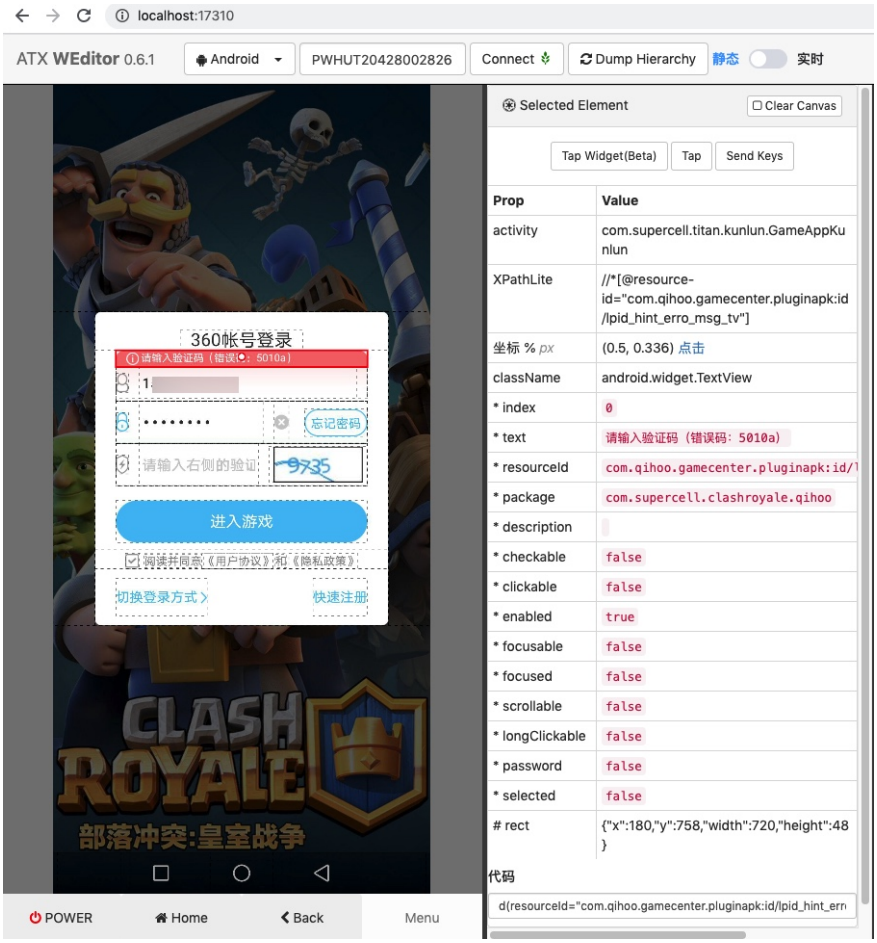


勾线后，点击 进入游戏

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：

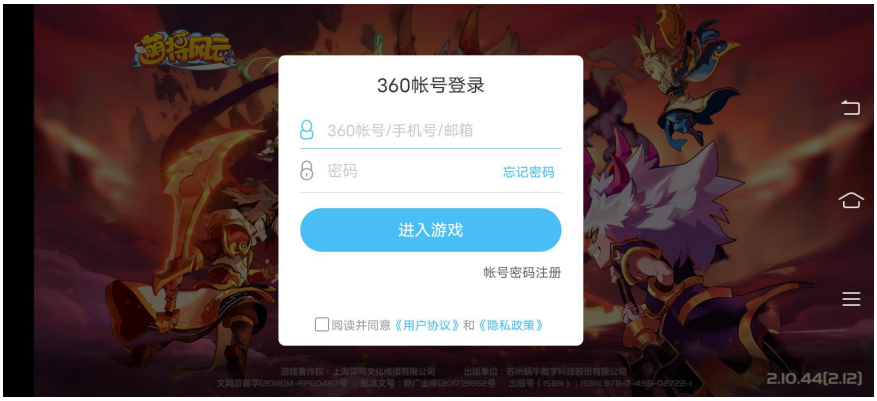
- `com.zhw.xzjh.qihoo_修真江湖`

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：

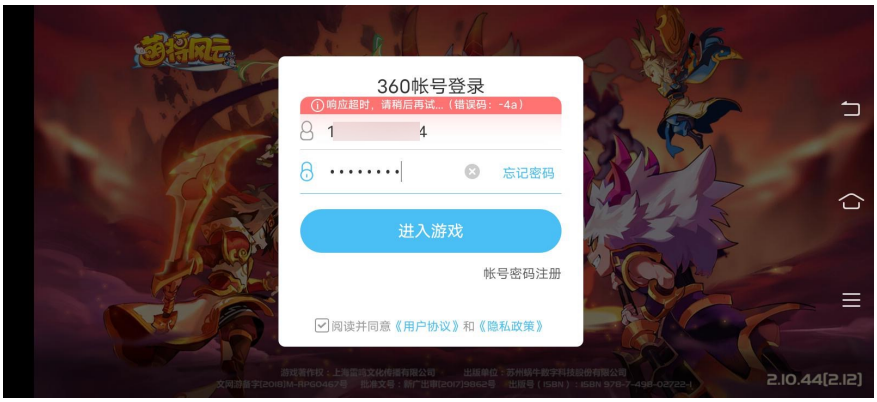
游戏：`com.qianhuan.mjfy.qihoo360/萌将风云`



-》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_剑侠棋缘`

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:

附录

下面列出相关参考资料。

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
-
- 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 - Airtest Project Docs
-