

目录

前言	1.1
简介	1.2
搭建环境	1.3
常见问题和心得	1.3.1
XCode编译WebDriverAgent.xcodeproj	1.3.1.1
开发心得	1.4
iOS的各种坑	1.5
获取源码xml	1.5.1
常用代码段	1.6
wda	1.6.1
crifan版wda	1.6.1.1
元素处理	1.6.2
iOS设备	1.6.3
app	1.6.4
app管理	1.6.4.1
首次登录引导页	1.6.4.2
微信	1.6.4.3
安全模式	1.6.4.3.1
微信公众号	1.6.4.3.2
设置	1.6.4.4
更新WiFi代理	1.6.4.4.1
AppStore	1.6.4.5
自动安装app	1.6.4.5.1
弹框处理	1.6.5
调试辅助	1.6.6
源码分析	1.7
附录	1.8
iOS内置app的bundle id	1.8.1
文档	1.8.2
参考资料	1.8.3

iOS自动化测试利器：facebook-wda

- 最新版本： v1.6
- 更新时间： 20210411

简介

总结iOS自动化主流框架facebook-wda的简介，环境搭建，包括安装依赖库、编译 WebDriverAgentRunner、安装WebDriverAgentRunner-Runner、启动test manager等，以及常见问题，包括USB端口转发、，以及开发的心得，尤其是iOS的各种坑，以及常用代码段，并给出部分源码分析，最后附上文档和参考资料。

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/ios_automation_facebook_wda: iOS自动化测试利器：facebook-wda](#)

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

- [iOS自动化测试利器：facebook-wda book.crifan.com](#)
- [iOS自动化测试利器：facebook-wda crifan.github.io](#)

离线下载阅读

- [iOS自动化测试利器：facebook-wda PDF](#)
- [iOS自动化测试利器：facebook-wda ePUB](#)
- [iOS自动化测试利器：facebook-wda Mobi](#)

版权说明

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

鸣谢

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

更多其他电子书

本人 crifan 还写了其他 100+ 本电子书教程，感兴趣可移步至：

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

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2021-04-11 11:46:47

简介

之前已在：

[移动端自动化测试概览](#)

中提到过，iOS自动化测试的主流框架之一是：[facebook-wda](#)

此处去详细介绍一下。

- [facebook-wda](#)
 - 主页
 - [openatx/facebook-wda: Facebook WebDriverAgent Python Client Library \(not official\)](#)
 - 作者：[openatx](#)
 - 语言：[Python](#)
 - 实现原理
 - 基于[Appium](#) 的 [WebDriverAgent](#)
 - 关于 [WebDriverAgent](#)
 - 简称： [WDA](#)
 - 是什么=一句话描述：一个基于[W3C](#) 的 [WebDriver](#) 的 [server](#) (的具体实现)
 - 底层依赖：[Apple](#) 的[XCUITest](#) (测试框架)
 - 起源和状态
 - 最早：[Facebook](#) 开发的
 - [Facebook的WebDriverAgent](#)
 - 现已暂停维护= [archived = read-only](#)
 - 主页
 - [facebookarchive/WebDriverAgent: A WebDriver server for iOS that runs inside the Simulator](#)
 - 现在：已停止维护
 - [Appium](#) 接手继续维护和更新
 - [Appium的WebDriverAgent](#)
 - 主页
 - [appium/WebDriverAgent: A WebDriver server for iOS that runs inside the Simulator](#)
 - 关于 [WebDriver](#)
 - 作者：[W3C](#)
 - 是什么：一套协议规范
 - 特点：与平台协议无关
 - 目的=作用：远程控制设备
 - 主页
 - <https://w3c.github.io/webdriver/>
 - 关于：[Apple](#) 的 [XCUITest](#)
 - 是什么：苹果的测试框架
 - 官网文档

- 之前：
 - [XCUITest](#)
 - (貌似) 最新
 - [XCTest | Apple Developer Documentation](#)
- 关于： [Appium](#)
 - [Appium: Mobile App Automation](#)

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2020-08-09 10:13:38

搭建环境

此处介绍如何用 `facebook-wda` 搭建iOS设备的自动化测试环境。

先介绍 `facebook-wda` 的架构流程图：

- 本地图片
 - 在线网页查看
 - [facebook-wda架构流程图](#)

开发环境概述

- 开发环境概述
 - `client` = 客户端
 - 你要测试的 `iOS` 设备，比如 `iPhone`
 - 给 `iPhone` 中安装 `WebDriverAgentRunner-Runner`
 - `server` = 服务端 = `test manager` = `WebDriverAgent`的服务
 - 需要在 `Mac` 中启动 `test manager`

首次：初始化

先介绍初始化需要做的事情，其中：

- 初始化 = 第一次 = 首次 = 只需要做一次，以后无需重复做

想要能自动化操作*iPhone*等*iOS*设备，需要

(1) 先确保Mac环境OK

把相关后续要用到的工具都安装好：

```
brew update
brew uninstall --ignore-dependencies libimobiledevice
brew uninstall --ignore-dependencies usbmuxd
brew install --HEAD usbmuxd
brew unlink usbmuxd
brew link usbmuxd
brew install --HEAD libimobiledevice
```

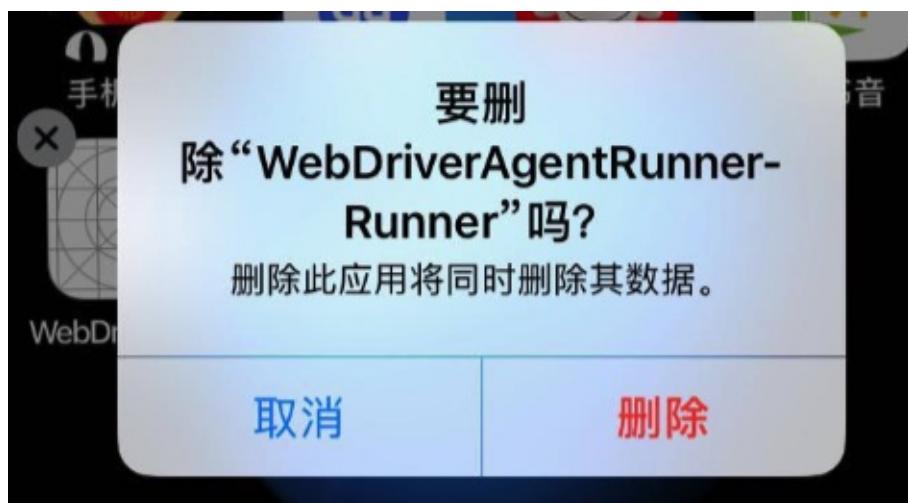
(2) 再去给 iPhone 中安装：

- 客户端 = APP = WebDriverAgentRunner-Runner
 - 用于配合 Mac 中的 server 端 的 test manager

安装后的效果：



此处长按待删除，才能看到app全名是： WebDriverAgentRunner-Runner：



给iPhone中安装WebDriverAgentRunner-Runner

核心思路，都是编译和安装app `WebDriverAgentRunner-Runner` 到iPhone中：

- 确保Mac中已安装XCode

下载代码：

```
git clone https://github.com/appium/WebDriverAgent.git
```

切换目录：

```
cd WebDriverAgent
```

看到相关文件：

```
11
total 96
-rw-r--r--  1 limao  CORP\Domain Users  2.0K  2 13 17:40 CONTRIBUTING.md
-rw-r--r--  1 limao  CORP\Domain Users  111B   2 13 17:40 Cartfile
-rw-r--r--  1 limao  CORP\Domain Users   75B   2 13 17:40 Cartfile.resolved
drwxr-xr-x  3 limao  CORP\Domain Users  96B   2 13 17:40 Configurations
drwxr-xr-x  4 limao  CORP\Domain Users 128B   2 13 17:40 Fastlane
-rw-r--r--  1 limao  CORP\Domain Users 177B   2 13 17:40 Gemfile
-rw-r--r--  1 limao  CORP\Domain Users  4.3K   2 13 17:40 Gemfile.lock
-rw-r--r--  1 limao  CORP\Domain Users  1.5K   2 13 17:40 LICENSE
-rw-r--r--  1 limao  CORP\Domain Users  1.9K   2 13 17:40 PATENTS
drwxr-xr-x  7 limao  CORP\Domain Users 224B   2 13 17:40 PrivateHeaders
-rw-r--r--  1 limao  CORP\Domain Users  2.5K   2 13 17:40 README.md
drwxr-xr-x  5 limao  CORP\Domain Users 160B   2 13 17:40 Scripts
drwxr-xr-x  5 limao  CORP\Domain Users 160B   2 13 17:40 WebDriverAgent.xcodeproj
drwxr-xr-x 16 limao  CORP\Domain Users 512B   2 13 17:40 WebDriverAgentLib
drwxr-xr-x  4 limao  CORP\Domain Users 128B   2 13 17:40 WebDriverAgentRunner
drwxr-xr-x  6 limao  CORP\Domain Users 192B   2 13 17:40 WebDriverAgentTests
drwxr-xr-x  5 limao  CORP\Domain Users 160B   2 13 17:40 ci-jobs
-rw-r--r--  1 limao  CORP\Domain Users 666B   2 13 17:40 gulpfile.js
-rw-r--r--  1 limao  CORP\Domain Users 896B   2 13 17:40 index.js
drwxr-xr-x  9 limao  CORP\Domain Users 288B   2 13 17:40 lib
-rw-r--r--  1 limao  CORP\Domain Users  2.5K   2 13 17:40 package.json
drwxr-xr-x  5 limao  CORP\Domain Users 160B   2 13 17:40 test
```

其中核心的入口文件，即Xcode项目文件是：`WebDriverAgent.xcodeproj`

关于如何编译和安装，则有2种方式：

- 通过IDE XCode 去编译和安装
 - Xcode -> Product -> Test
- 在 终端 运行 `xcodebuild` 命令去编译和安装
 - Terminal 中：运行 `xcodebuild` 的 `test`

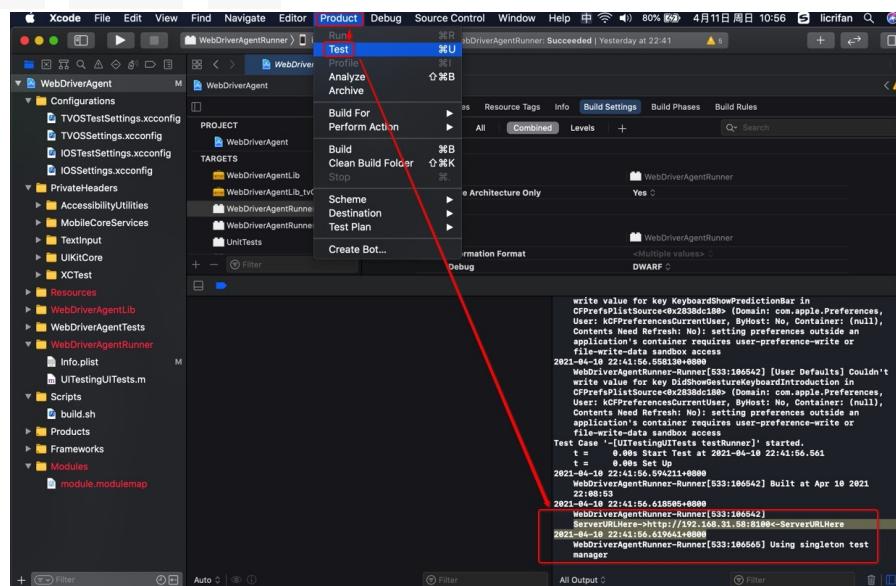
上述操作步骤，和后续的每次运行 test manager 的方式是一样的，所以细节放在后面介绍。

之后：每次调试之前

启动test manager服务

- server = 服务端 = test manager = WebDriverAgent的服务
 - 需要在 Mac 中启动 test manager
 - 2种方式
 - XCode

■ Xcode -> Product -> Test



■ 终端

- Terminal 中: 运行 xcdebuild 的 test
- 直接一步:

```
xcdebuild -project WebDriverAgent.xcodeproj -scheme WebDriverAgentRunner -destination "id=$idevice_id -1 | head -n1" test
```

■ 或分2步

- 先获取iOS设备的UDID:

```
CUR_UDID=$(idevice_id -1 | head -n1)
```

- 再运行:

```
xcdebuild -project WebDriverAgent.xcodeproj -scheme WebDriverAgentRunner -destination "id=$CUR_UDID" test
```

■ 注:

- 要在 WebDriverAgent 的目录中运行上述命令
- idevice_id -1 作用是列出当前连接到Mac中的所有iOS的设备（的

- UDID)
 - 详见: `idevice_id`
 - `head -n1` 作用是获取第一个 (iOS设备的UDID)

最后能看到输出 `ServerURLHere` 和 `Using singleton test manager` :

```
...
Test Case '-[UITestingUITests testRunner]' started.
t = 0.01s Start Test at 2020-02-20 10:50:59.818
t = 0.01s Set Up
2020-02-20 10:50:59.968359+0800 WebDriverAgentRunner-Runner[460:142725] Built at Feb 20 20
20 10:50:08
2020-02-20 10:51:00.119667+0800 WebDriverAgentRunner-Runner[460:142725] ServerURLHere- ht
tp://192.168.31.43:8100 -ServerURLHere
2020-02-20 10:51:00.123946+0800 WebDriverAgentRunner-Runner[460:142853] Using singleton te
st manager
```

即表示正常启动了 `test manager = WDA的server` 了。

如何确认 `test manager` 服务已正常运行

可以去访问运行了 `test manager` 最后所输出的地址:

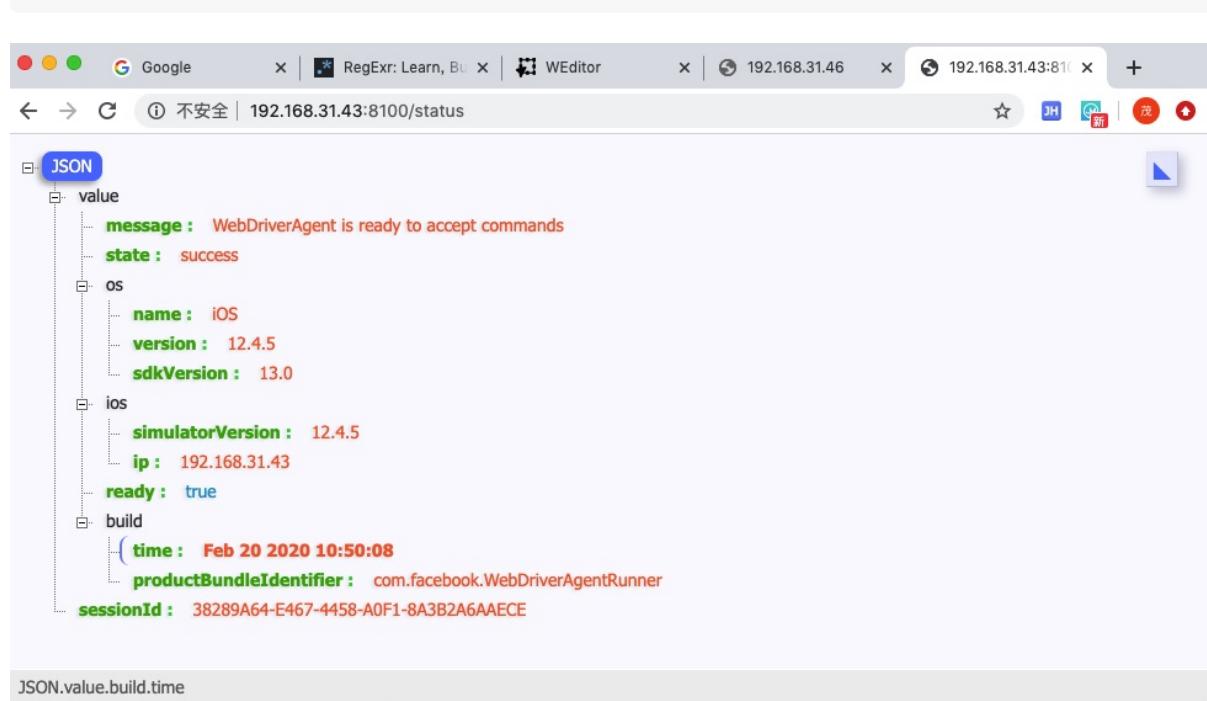
`http://192.168.31.43:8100`

加上 `status` 后是:

`http://192.168.31.43:8100/status`

会输出当前状态信息:

```
{
  "value": {
    "message": "WebDriverAgent is ready to accept commands",
    "state": "success",
    "os": {
      "name": "iOS",
      "version": "12.4.5",
      "sdkVersion": "13.0"
    },
    "ios": {
      "simulatorVersion": "12.4.5",
      "ip": "192.168.31.43"
    },
    "ready": true,
    "build": {
      "time": "Feb 20 2020 10:50:08",
      "productBundleIdentifier": "com.facebook.WebDriverAgentRunner"
    }
  },
  "sessionId": "38289A64-E467-4458-A0F1-8A3B2A6AAECE"
}
```



```
JSON
value
  message : WebDriverAgent is ready to accept commands
  state : success
  os
    name : iOS
    version : 12.4.5
    sdkVersion : 13.0
  ios
    simulatorVersion : 12.4.5
    ip : 192.168.31.43
    ready : true
  build
    time : Feb 20 2020 10:50:08
    productBundleIdentifier : com.facebook.WebDriverAgentRunner
    sessionId : 38289A64-E467-4458-A0F1-8A3B2A6AAECE

JSON.value.build.time
```

USB端口转发

为了测试更方便，最好安装和启动端口转发

以及，如果通过IP地址访问：`http://192.168.31.43:8100/status`，无法显示，则可以，换用端口转发

具体方式是，用 `iproxy` 或 `mobiledevice` 实现，把访问Mac本地的端口，转发到USB连接着的iOS设备中

命令：

对于只连接单个iOS设备，比如某个iPhone的话，只需要：

```
iproxy 8100 8100
```

或：

```
mobiledevice tunnel 8100 8100
```

更多解释和用法，详见：

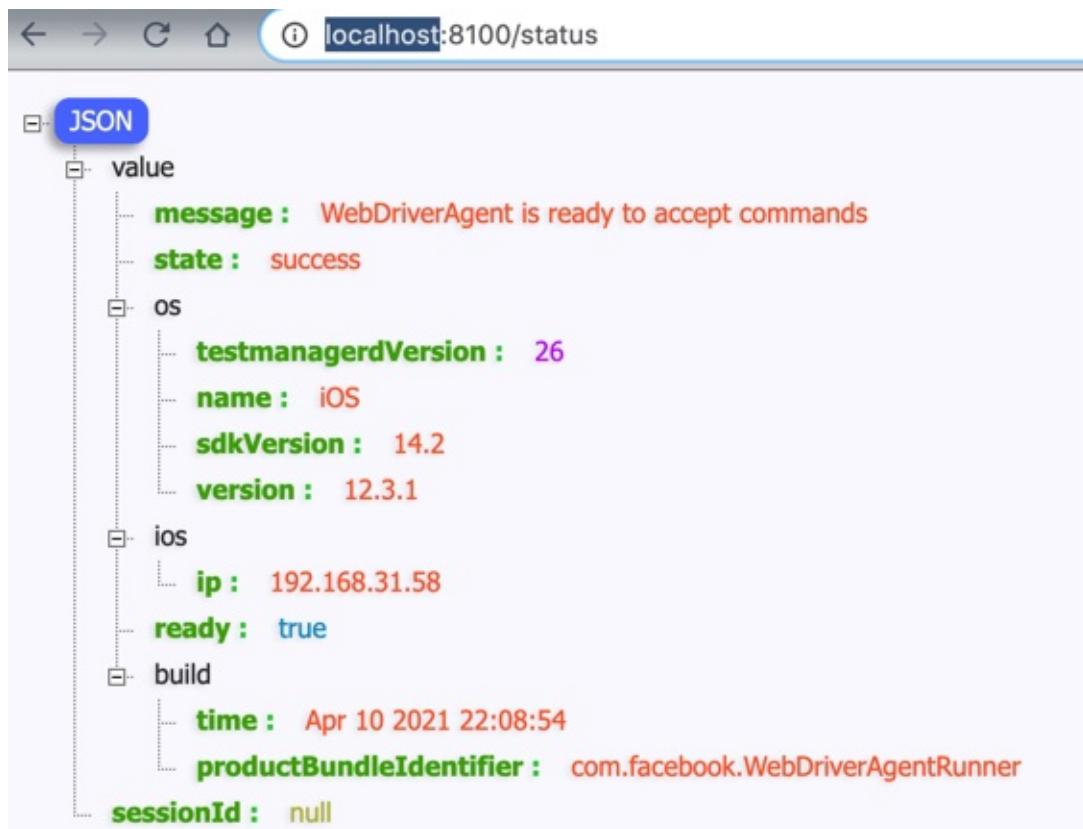
[端口转发 · 苹果相关开发总结](#)

如果已端口转发则可以把IP换localhost

此时 `status` 的地址就是，把IP换成localhost：

```
http://localhost:8100/status
```

即可正常显示，返回状态信息：



```
localhost:8100/status
```

JSON

```
{  
  "value": {  
    "message": "WebDriverAgent is ready to accept commands",  
    "state": "success",  
    "OS": {  
      "testmanagerdVersion": 26,  
      "name": "iOS",  
      "sdkVersion": 14.2,  
      "version": "12.3.1"  
    },  
    "ios": {  
      "ip": "192.168.31.58",  
      "ready": true  
    },  
    "build": {  
      "time": "Apr 10 2021 22:08:54",  
      "productBundleIdentifier": "com.facebook.WebDriverAgentRunner",  
      "sessionId": null  
    }  
  }  
}
```

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2021-04-11 11:03:35

搭建环境期间常见问题和心得

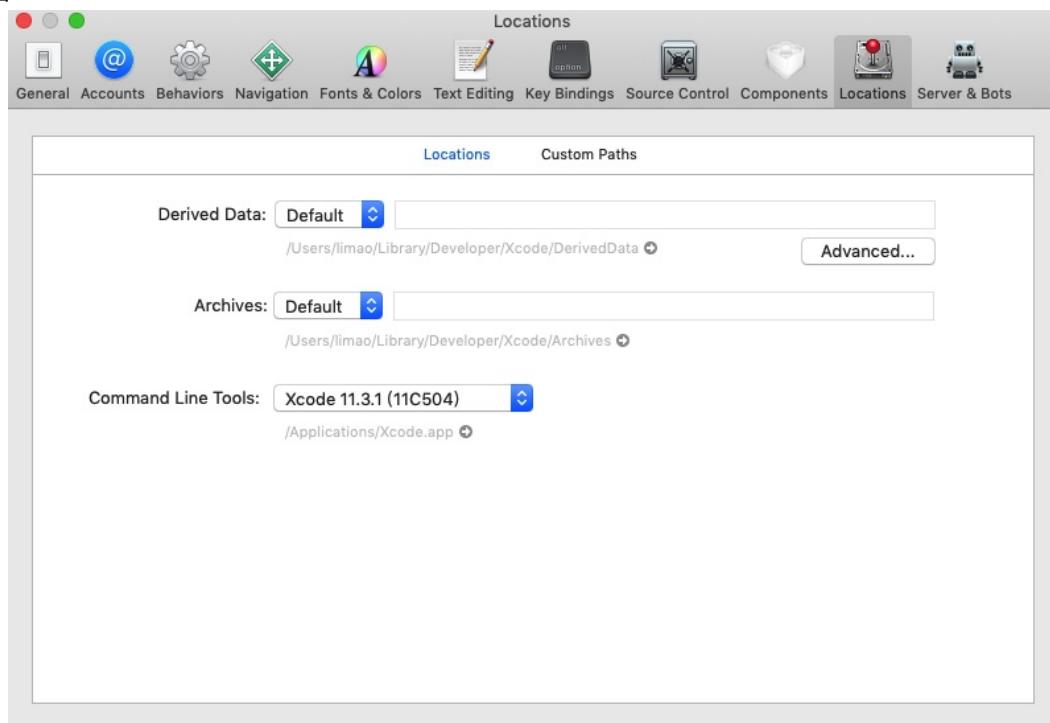
下面整理一些搭建环境期间的常见问题和心得总结：

xcodebuild报错：xcode-select error tool xcodebuild requires Xcode

如果运行xcodebuld报错：

```
xcode-select: error: tool 'xcodebuild' requires Xcode, but active developer directory '/Library/Developer/CommandLineTools' is a command line tools instance
```

- 原因：没有安装XCode 或 虽然已安装XCode，但是没启用XCode的命令行
- 解决办法：去安装并开启XCode的命令行
- 步骤：
 - 文字
 - Xcode -> 设置 -> Locations -> Command Line Tools， 默认是空，下拉选择 Xcode 11.3.1(11C504)
 - 截图



安装后，即可查看版本信息：

```
~ xcodebuild -version
Xcode 11.3.1
Build version 11C504
```

xcodebuild报错：xcodebuild error missing value for key

如果没有iOS设备（如iPhone）插入到Mac中，则运行：

```
xcodebuild -project WebDriverAgent.xcodeproj -scheme WebDriverAgentRunner -destination "id=iDevice_id -l | head -n1" test
```

会报错：

```
~/dev/xxx/crawler/appAutoCrawler/AppCrawler/iOSAutomation/refer(WebDriverAgent master •
  xcodebuild -project WebDriverAgent.xcodeproj -scheme WebDriverAgentRunner -destination
  "id=iDevice_id -l | head -n1" test
  xcodebuild: error: missing value for key 'id' of option 'Destination'
```

当前被测iOS设备详情

在启动 `test manager` 期间会输出当前被测设备的详细信息

举例：

(1) iOS 12.4.5 的 iPhone6

```
2020-05-07 09:20:31.198 xcodebuild[2440:2434041] [MT] IDETestOperationsObserverDebug: (B79
57682-E70F-46C7-86C2-53AEE7C8993D) Beginning test session WebDriverAgentRunner-B7957682-E7
0F-46C7-86C2-53AEE7C8993D at 2020-05-07 09:20:31.194 with Xcode 11C504 on target DVTiOS
Device (0x7f8456759e30), Crifan iPhone6, iPhone, 12.4.5 (16G161), ed94089f3e34d5538065a695
bfdf03dfbb3c5579 {
    deviceSerialNumber: DNPND9S1G5MR
    identifier: ed94089f3e34d5538065a695bfdf03dfbb3c5579
    deviceClass: iPhone
    deviceName: Crifan iPhone6
    deviceIdentifier: ed94089f3e34d5538065a695bfdf03dfbb3c5579
    productVersion: 12.4.5
    buildVersion: 16G161
    deviceSoftwareVersion: 12.4.5 (16G161)
    deviceArchitecture: arm64
    deviceTotalCapacity: 60058931200
    deviceAvailableCapacity: 38391648256
    deviceIsTransient: NO
    ignored: NO
    deviceIsBusy: NO
    deviceIsPaired: YES
    deviceIsActive: YES
    deviceActivationState: Activated
    isPasscodeLocked: NO
    deviceType: DVTDeviceType:0x7f845621d390 Xcode.DeviceType.iPhone
    supportedDeviceFamilies: ()
```

```
)  
    applications:          (null)  
    provisioningProfiles: (null)  
    hasInternalSupport:   NO  
    hasWritableSystem:    NO  
    isSupportedOS:        YES  
    bootArgs:             (null)  
    nextBootArgs:          (null)  
    connected:            YES  
    isWirelessEnabled:    NO  
    connectionType:       direct  
    hostname:             (null)  
    bonjourServiceName:   d4:f4:6f:0a:30:80@fe80::d6f4:6fff:fe0a:3080._apple-mobd  
    ev2._tcp.local.  
    activeProxiedDevice:  (null)  
} (12.4.5 (16G161))
```

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2021-04-11 11:43:02

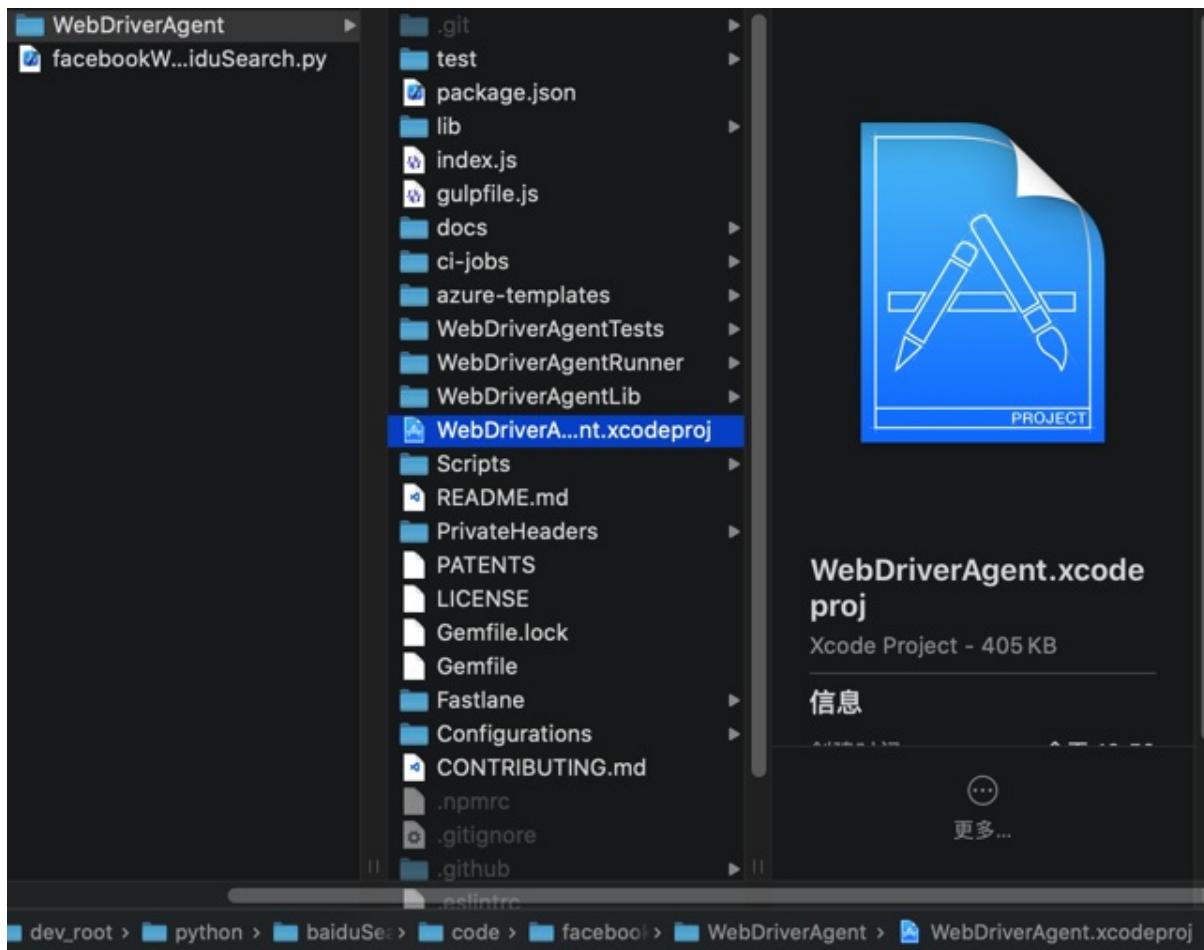
如何用XCode编译 WebDriverAgent.xcodeproj

对于下载到 WebDriverAgent 的源码中的 WebDriverAgent.xcodeproj , 第一次编译最好去用XCode编译。

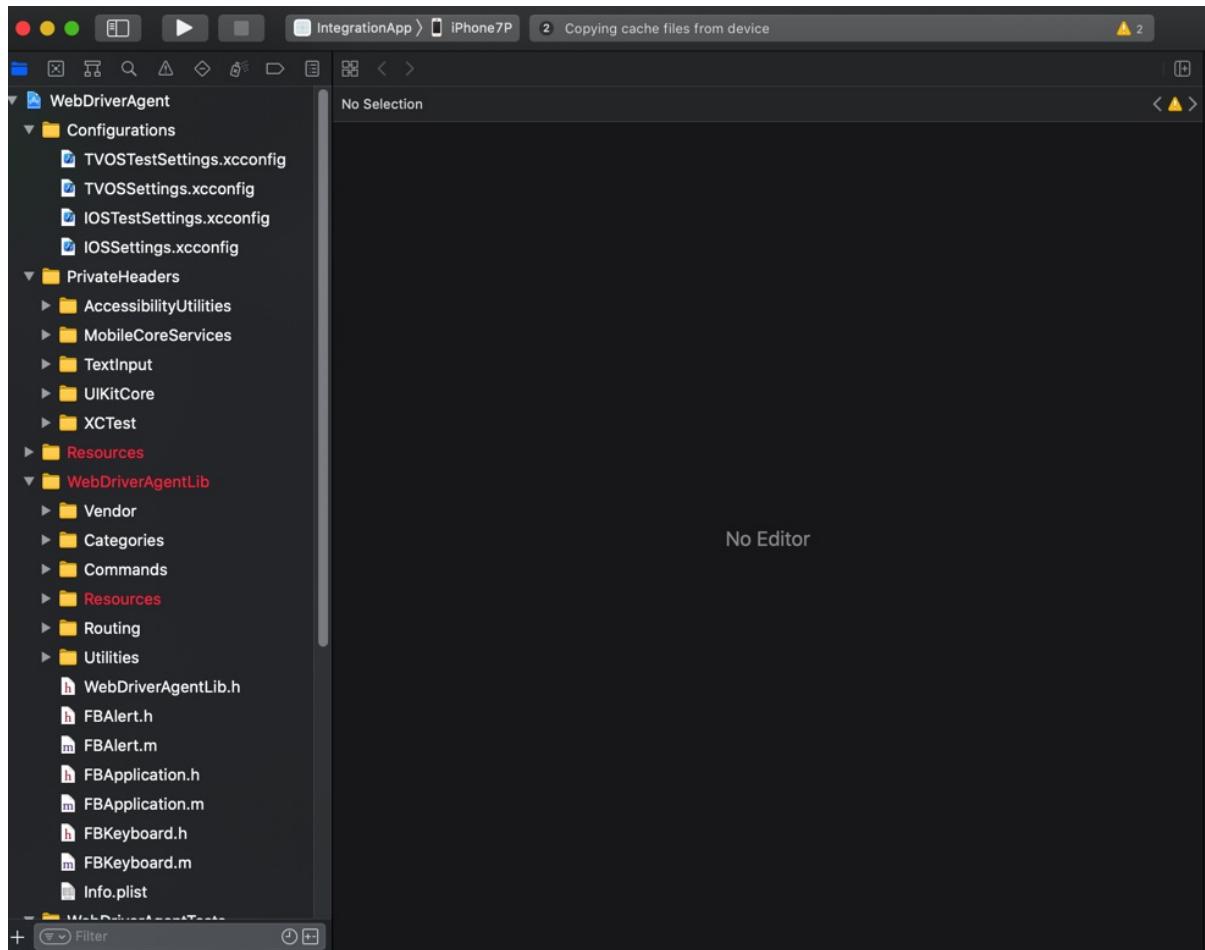
因为往往涉及到配置 Team 和 自动签名 等事宜。

下面就来介绍，如何用 XCode 去配置和编译 WebDriverAgent.xcodeproj

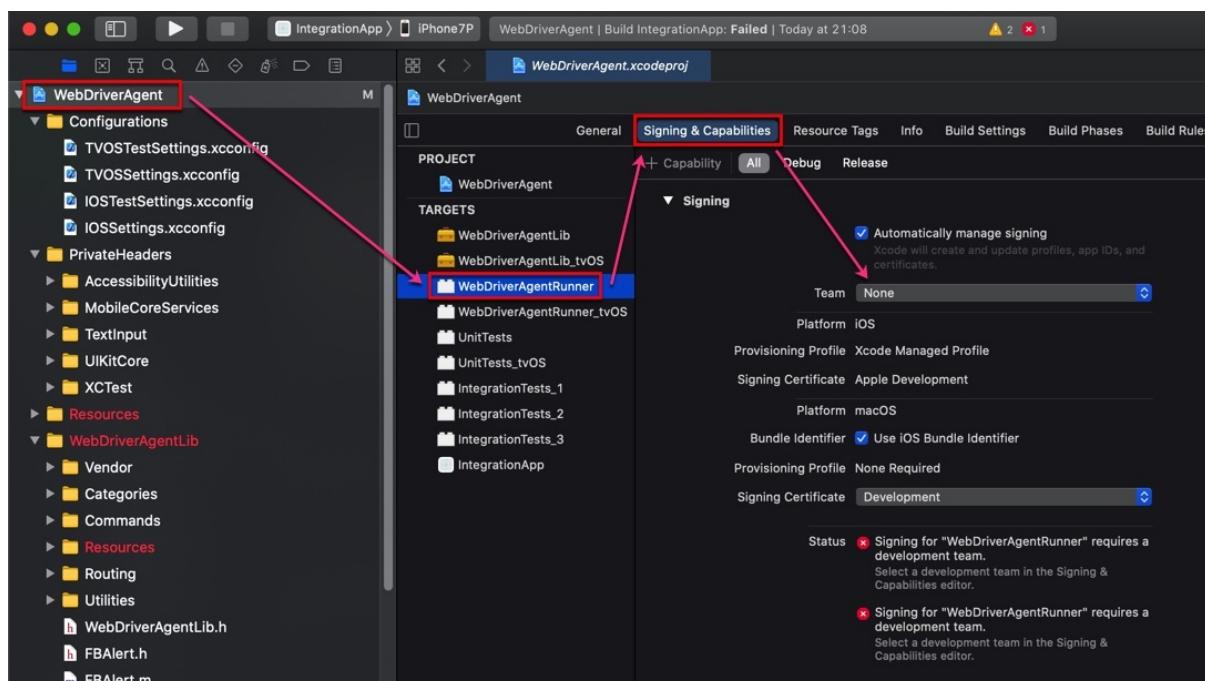
双击 WebDriverAgent.xcodeproj



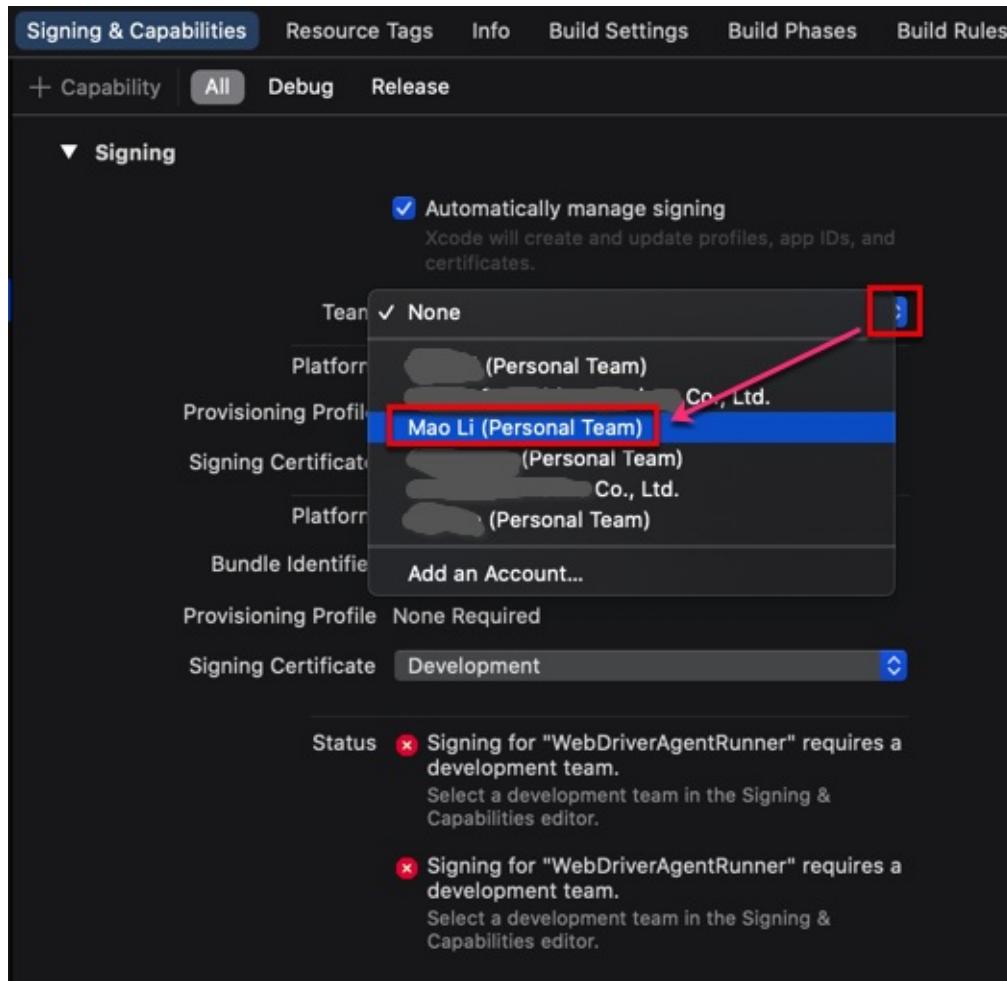
会自动用XCode打开：



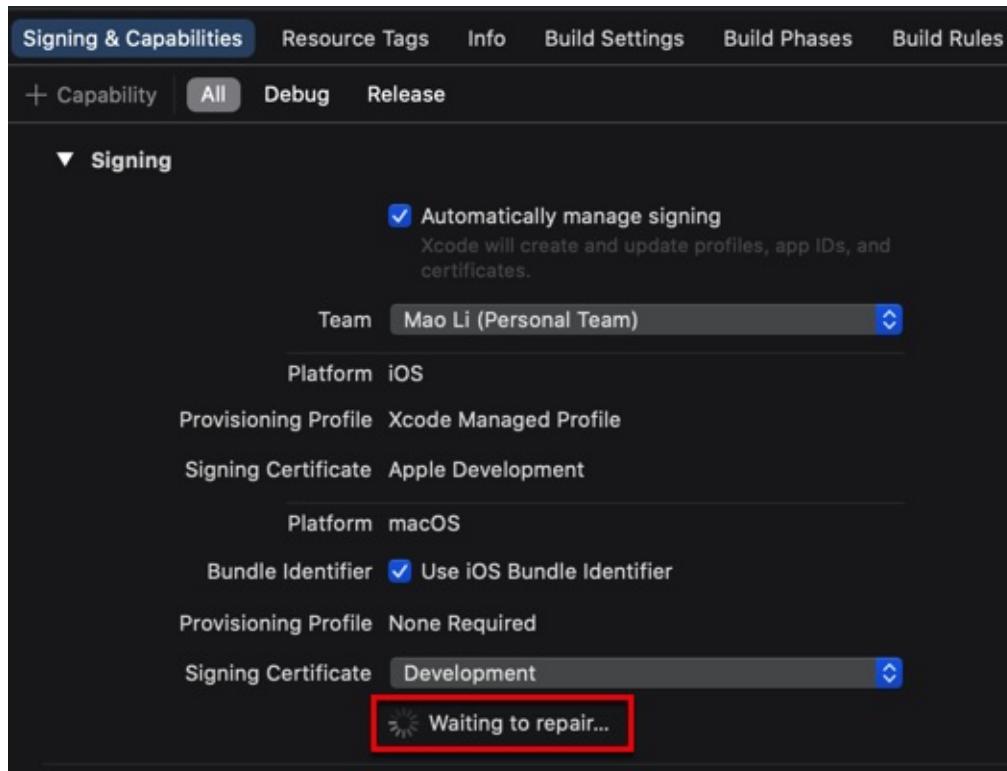
点击左上角的项目，进入项目属性，点击 TARGETS 中的 WebDriverAgentRunner，切换到 Signing & Capabilities：



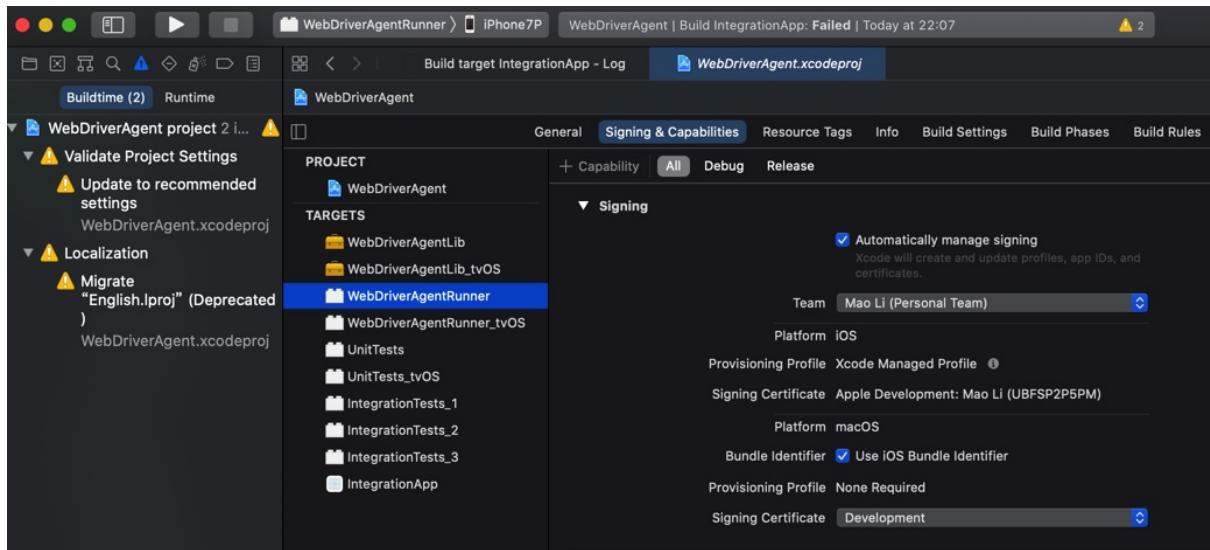
默认 Team 是 None，需要去选择一个自己的苹果账号：



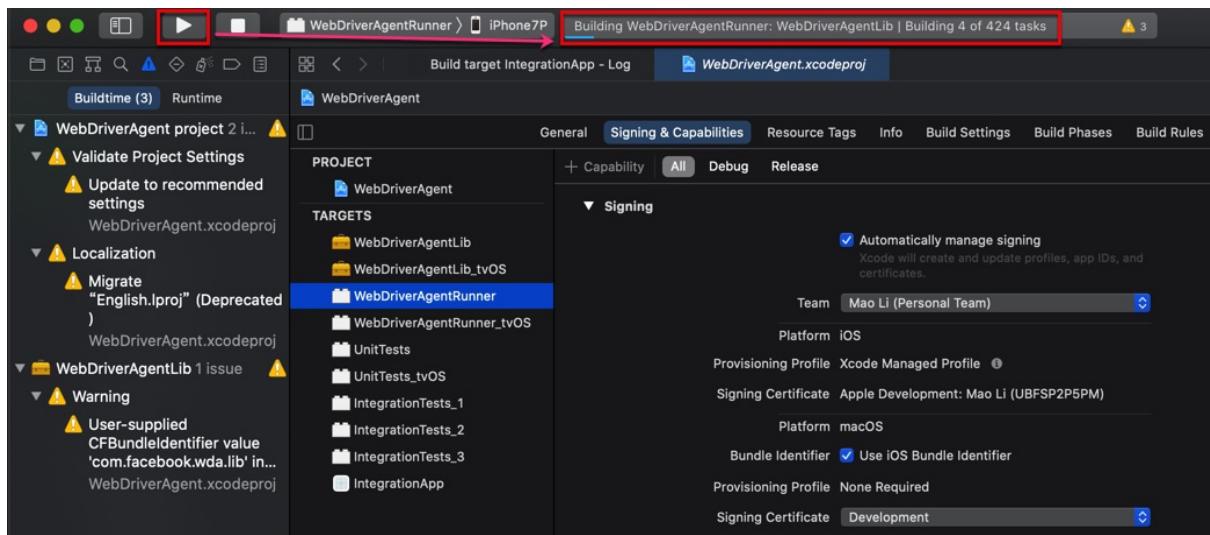
然后会触发自动修复，显示 Waiting to repair :



看到没有其他警告或错误，就表示自动创建签名和Profile等工作正常了：



接着即可去编译了：点击左上角 ▶ 按钮，即可触发编译，显示 Building ...

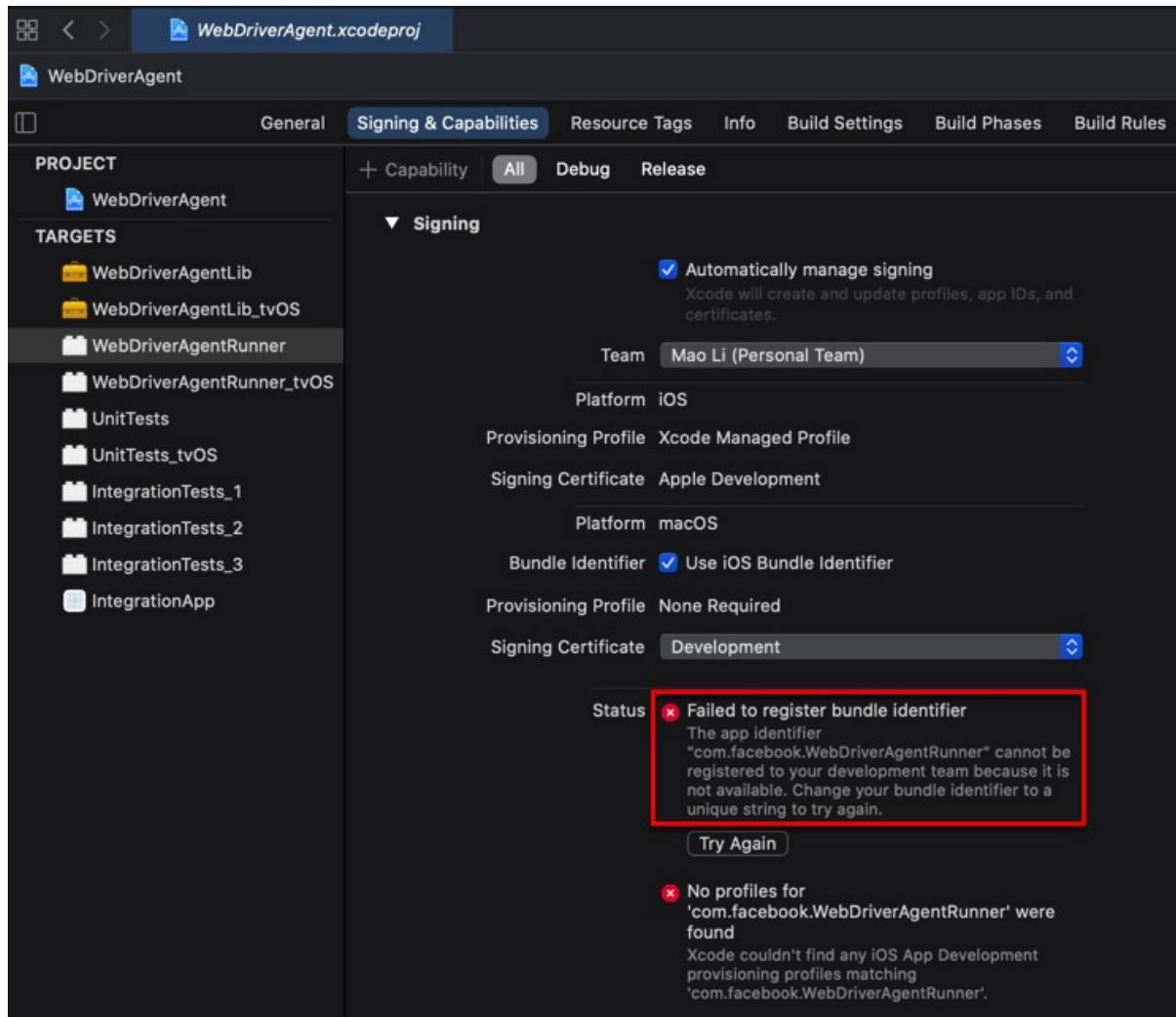


之后即可正常的 Product -> Test 去测试，启动服务，供后续使用了。

Failed to register bundle identifier

如果 Signing & Capabilities 的自动修复后报错：

```
Failed to register bundle identifier
The app identifier "com.facebook.WebDriverAgentRunner" cannot be registered to your development team because it is not available. Change your bundle identifier to a unique string to try again.
```

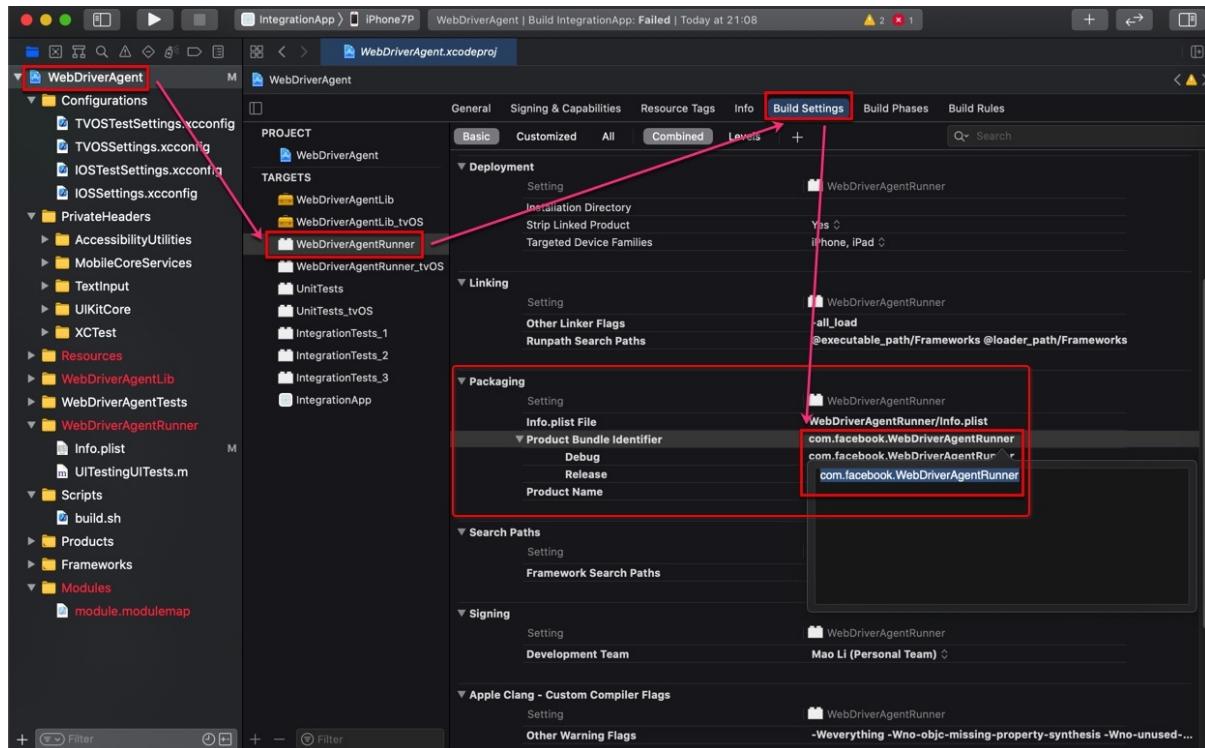


原因：（很可能是）默认的ID：`com.facebook.WebDriverAgentRunner` 已存在，重复了，导致无法继续。

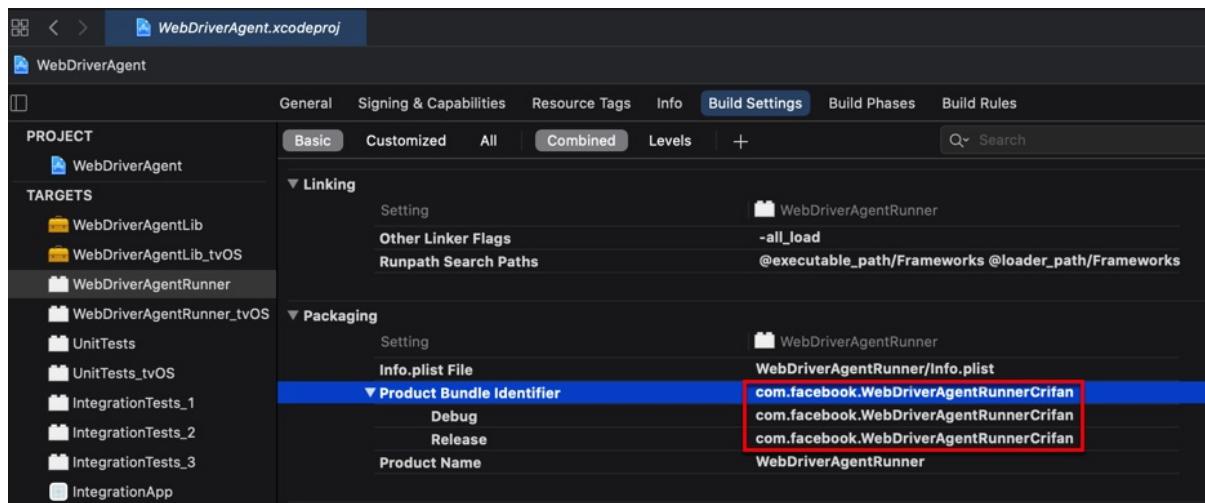
解决办法：修改为其他（独一无二的）值

操作步骤：

`WebDriverAgentRunner` 的属性 -> `Build Settings` -> `Packaging` -> `Product Bundle Identifier`



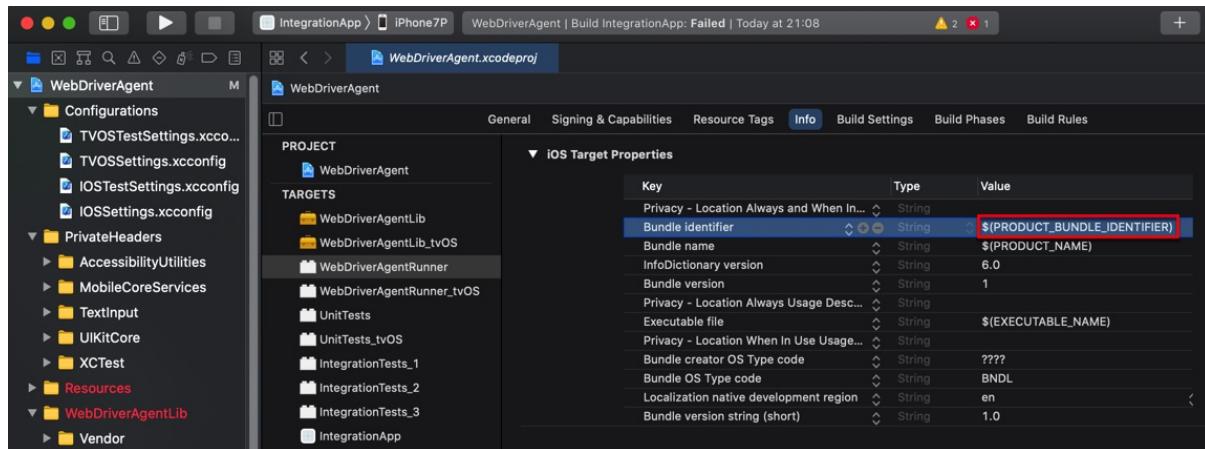
把值从默认的： com.facebook.WebDriverAgentRunner 改为别的，确保不重复的值，比如我此处改
为： com.facebook.WebDriverAgentRunnerCrifan



别处调用到此处的Product Bundle Identifier

后来注意到一个细节，别处会调用到此处的 Product Bundle Identifier 中的值

比如： Info -> Key -> Bundle Identifier : \$(PRODUCT_BUNDLE_IDENTIFIER)



A build only device cannot be used to run this target

如果编译期间报错：

```
A build only device cannot be used to run this target.
No supported iOS devices are available. Connect a device to run your application or choose
a simulated device as the destination.
```

则说明是没有选择正确的设备。

此处是需要去确保iOS设备已连接，并选择目标设备为对应iOS设备。

此处iOS设备是： iPhone7P

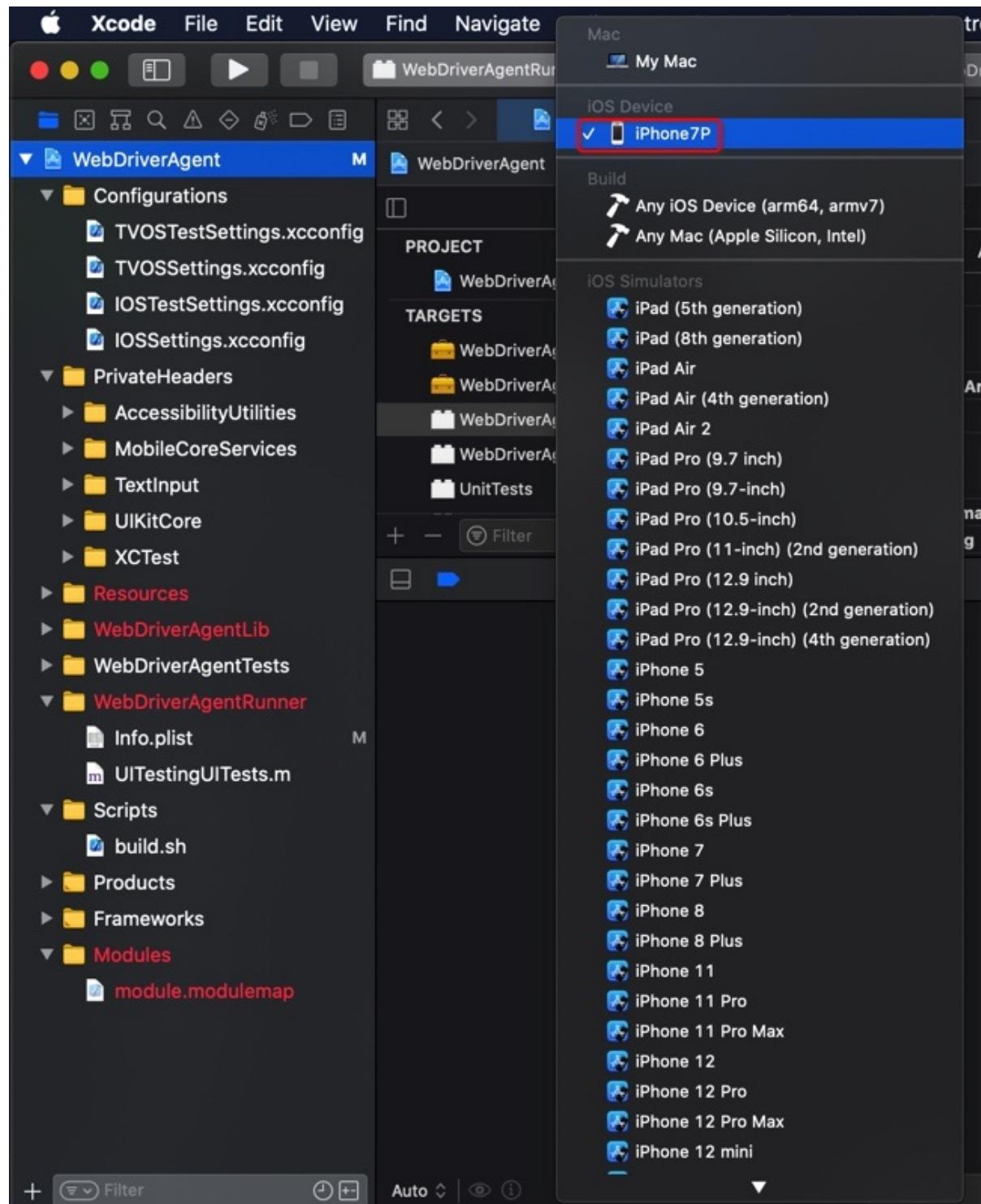
先连接到Mac中：



以及可以借助于 `iDevice_id -l` 去确保能找到设备ID，说明设备的确已连接：

```
iDevice_id -l
3dc13714e21415898e8e2c2863d96990a4d69c97
```

然后去选择目标设备为 iPhone7P



后续即可正常编译和测试。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2021-04-11 11:44:01

开发心得

wda如何同时测试多个设备

- 问：如何用wda同时测试多个设备？
- 答：使用不同端口转发

具体做法举例：

```
iproxy 8100 8100
iproxy 8101 8101
iproxy 8102 8102
```

代码中，本地连接不同端口：

```
gWdaClient0 = wda.Client('http://localhost:8100')
gWdaClient1 = wda.Client('http://localhost:8101')
gWdaClient2 = wda.Client('http://localhost:8102')
```

即可。

感慨：对于apple的态度

见到[别人](#)有提到：

Apple公司因其无与伦比的设计，让无数果粉为之迷恋

但作为iOS测试人员，也因为iOS系统封闭和不开放苦不堪言，羡慕死Android测试

对此深有体会，不能再同意更多：

- 消费者：对于apple产品觉得很好看，很喜欢
- 测试、自动化人员：苦不堪言
 - 原因：apple生态封闭，不开放
 - 虽然提供了XCTest，但是很不好用
 - iOS, Mac等内部的库是不开放的
 - 没法直接用来做测试和自动化
 - -》 Facebook的WebDriverAgent（后由Appium维护）已经做到了
 - 用工具从私有的库中dump出头文件和api接口
 - 但是实际用起来，仍然是各种bug和不兼容
 - 包括但不限于（后续会介绍到的）[各种坑](#)
 - 获取不到源码
 - 只能获取部分源码
 - 获取源码会导致test manager崩溃（需要重装 WebDriverAgentRunner）

- 无法完美支持元素visible属性
- 获取到的源码很混乱
 - 比如
 - 包含了前一页（甚至几页）的xml源码

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2021-04-11 11:45:49

iOS的各种坑

在用 `facebook-wda + WebDriverAgent` 去自动化测试iOS设备期间，遇到各种坑。

其中多数坑，都是苹果官方的API不稳定或有bug导致的，少数是 `WebDriverAgent` 或 `facebook-wda` 的。

wda找到元素，点击元素，竟然偶尔会无效

对于页面：



代码去找到并点击 个人所得税：

```
isIntoDetailOk = CommonUtils.multipleRetry(
    {
        "functionCallback": self.appStoreSearchResultIntoDetail,
        "functionParaDict": {
            "appName": appName,
        }
    },
    maxRetryNum = 10,
    sleepInterval = 0.5,
```

```

    )

def appStoreSearchResultIntoDetail(self, appName):
    """for AppStore search result list page
       try find first match result
       then click into detail page

    Args:
        appName (str): app name
    Returns:
        bool, dict
            bool: is into detail page or not
    Raises:
    """
    isIntoDetailOk = False
    """

    搜索结果列表页 京东 重新下载:
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
    " x="0" y="0" width="414" height="736">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
        " x="0" y="0" width="414" height="736">
            <XCUIElementTypeCollectionView type="XCUIElementTypeCollectionView" en
            abled="true" visible="true" x="0" y="0" width="414" height="736">
                <XCUIElementTypeCell type="XCUIElementTypeCell" name="京东-一起热爱
                就现在，新人送188元购物礼包，四颗半星，34.6万 个评分" label="京东-一起热爱 就现在，新人送188元购
                物礼包，四颗半星，34.6万 个评分" enabled="true" visible="true" x="20" y="86" width="374" hei
                ght="295">
                    <XCUIElementTypeButton type="XCUIElementTypeButton" name="重新
                    下载" label="重新下载" enabled="true" visible="true" x="364" y="86" width="30" height="68"/>
                </XCUIElementTypeCell>
                <XCUIElementTypeCell type="XCUIElementTypeCell" name="京东金融-超过
                三亿人在用，享24期免息，赚20亿金贴，四颗半星，20.8万 个评分" label="京东金融-超过三亿人在用，享24
                期免息，赚20亿金贴，四颗半星，20.8万 个评分" enabled="true" visible="true" x="20" y="420" wid
                th="374" height="295">
                    <XCUIElementTypeButton type="XCUIElementTypeButton" name="打开"
                    label="打开" enabled="true" visible="true" x="320" y="420" width="74" height="69"/>
                </XCUIElementTypeCell>
            </XCUIElementTypeCollectionView>
        </XCUIElementTypeOther>
    </XCUIElementTypeOther>

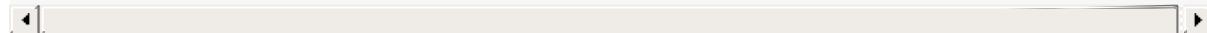
    搜索结果列表页 美团 获取:
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
    " x="0" y="0" width="414" height="736">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
        true" x="0" y="0" width="414" height="736">
            <XCUIElementTypeCollectionView type="XCUIElementTypeCollectionView" en
            abled="true" visible="true" x="0" y="0" width="414" height="736">
                <XCUIElementTypeCell type="XCUIElementTypeCell" name="美团-吃喝玩乐
                尽在美团，美食外卖买菜买药首选美团App，四又四分之三颗星，94.7万 个评分" label="美团-吃喝玩乐 尽在
                美团，美食外卖买菜买药首选美团App，四又四分之三颗星，94.7万 个评分" enabled="true" visible="true"
                " x="20" y="86" width="374" height="295">
                    <XCUIElementTypeButton type="XCUIElementTypeButton" name="获取"

```

```

label="获取" enabled="true" visible="true" x="320" y="86" width="74" height="68"/>
    </XCUIElementTypeCell>
    <XCUIElementTypeCell type="XCUIElementTypeCell" name="美团外卖-外卖
订餐,送啥都快, 美食生鲜,水果下午茶, 四又四分之三颗星, 607万 个评分" label="美团外卖-外卖订餐,送啥
都快, 美食生鲜,水果下午茶, 四又四分之三颗星, 607万 个评分" enabled="true" visible="true" x="20"
y="420" width="374" height="295">
        <XCUIElementTypeButton type="XCUIElementTypeButton" name="获取"
label="获取" enabled="true" visible="true" x="320" y="420" width="74" height="69"/>
        </XCUIElementTypeCell>
        </XCUIElementTypeCollectionView>
    </XCUIElementTypeOther>
</XCUIElementTypeOther>
"""
parentCollectionViewClassChain = "/XCUIElementTypeCollectionView[`rect.x = 0 AND rect.
y = 0 AND rect.width = %d AND rect.height = %d`]" % (self.X, self.totalY)
firstMatchCellQuery = {"type": "XCUIElementTypeCell", "nameContains": appName, "enabled": "true"}
firstMatchCellQuery["parent_class_chains"] = [parentCollectionViewClassChain]
foundAndClicked = self.findAndClickElement(query=firstMatchCellQuery, timeout=0.2)
isIntoDetailOk = foundAndClicked
return isIntoDetailOk

```



始终都是正常的：可以找到并点击元素，然后会进入app下载的详情页

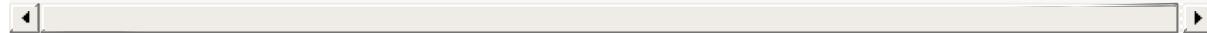
但是目前调试期间，先后遇到2次了

只是对于特殊的app名字： 个人所得税 ，出现了虽然代码中能找到元素，并点击了元素：

```

[200609 15:45:29][DevicesMethods.py 851] True to Clicked element {'type': 'XCUIElementType
Cell', 'nameContains': '个人所得税', 'enabled': 'true', 'parent_class_chains': ['/XCUIEleme
ntTypeCollectionView[`rect.x = 0 AND rect.y = 0 AND rect.width = 414 AND rect.height = 736`]
']}

```



但是实际上：

竟然点击没生效

-» 页面没有进入后续的详情页

-» 但是同样代码，重新测试，却又正常，可以点击元素进入详情页了：



很是诡异。

根本原因：至今未知。

暂时在列表页前后加上等待时间：

```
# Special: try add some wait time to avoid some special case:
# for 个人所得税 search result page, found and click 个人所得税, but sometime actually not into detail page
time.sleep(0.5)
isInDetailOk = CommonUtils.multipleRetry(
    {
        "functionCallback": self.appStoreSearchResultIntoDetail,
        "functionParaDict": {
            "appName": appName,
        }
    },
)
```

```
    maxRetryNum = 10,
    sleepInterval = 0.5,
)
if not isIntoDetailOk:
    respInfo = "Fail to into app detail page for %s" % appName
    return isInstallOk, respInfo
# Special: try add some wait time to avoid some special case:
# for 个人所得税 search result page, found and click 个人所得税, but sometime actually not i
nto detail page
time.sleep(0.2)
```

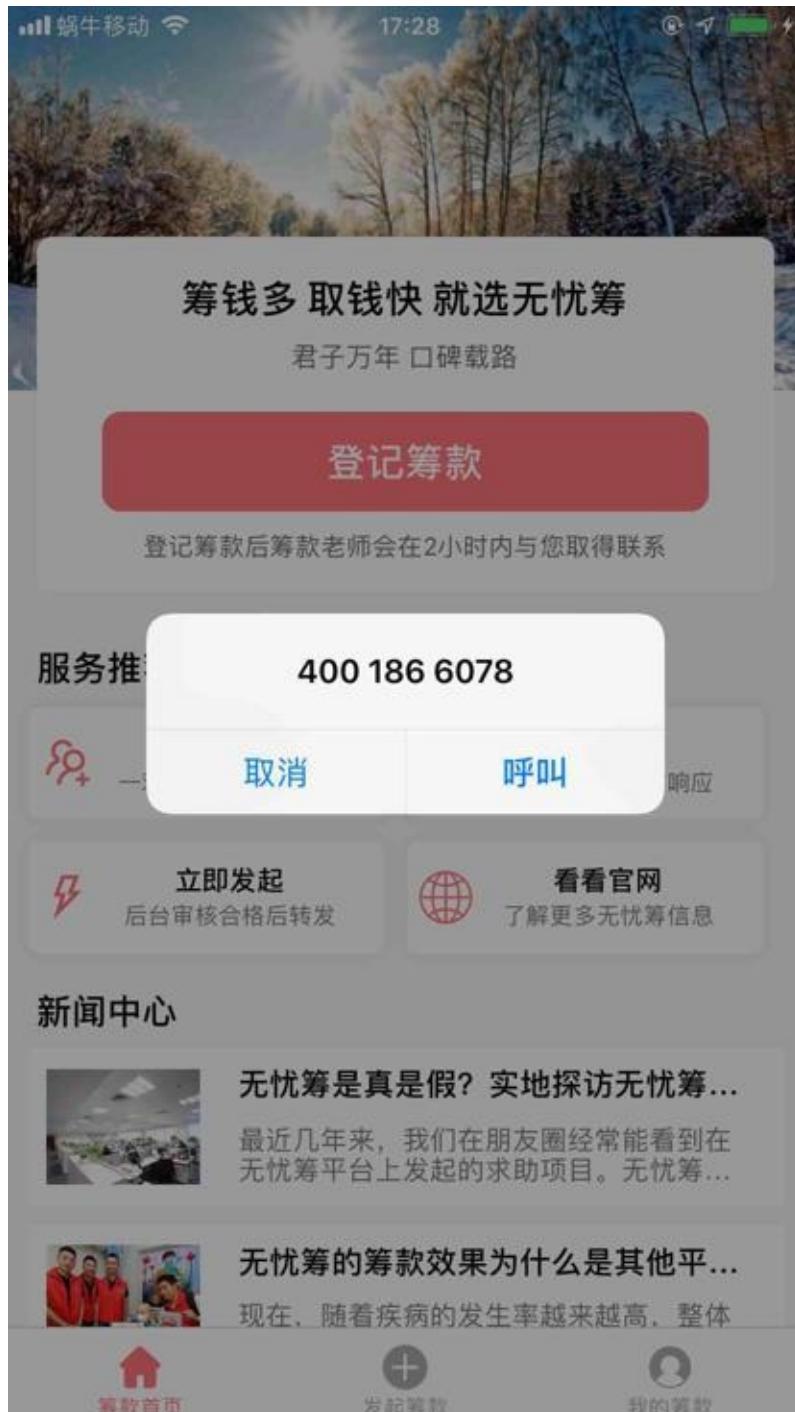
希望，或许能稳定些，或许能规避此问题？

详见：

【未解决】facebook-wda点击个人所得税元素无效：没有进入AppStore详情页

偶尔会遇到 通过坐标值点击元素 无效 实际上误点击别的位置

对于页面：



代码已找到了取消按钮，然后去点击 其中间坐标位置

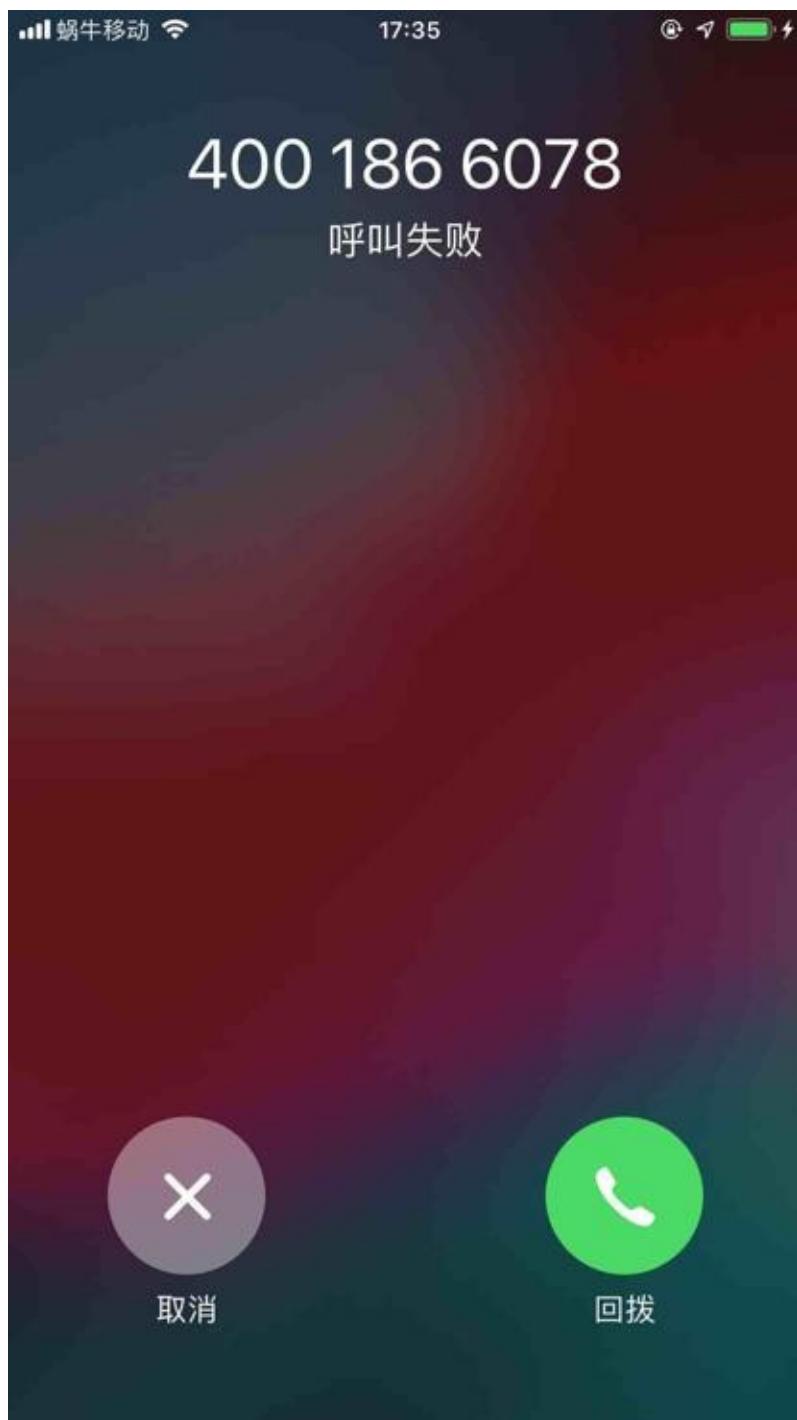
```
clickCenterPosition(curSession, cancelSoup.attrs)

def clickCenterPosition(curSession, elementAttrDict):
    x = int(elementAttrDict["x"])
    y = int(elementAttrDict["y"])
    width = int(elementAttrDict["width"])
    height = int(elementAttrDict["height"])
    centerX = x + int(width / 2)
    centerY = y + int(height / 2)
```

```
curSession.click(centerX, centerY)
logging.info("Clicked [%s, %s]", centerX, centerY)
```

之前此点击元素中间位置的代码工作都是正常的

唯独这此，点击 取消按钮 后，实际上是点击了：呼叫 按钮的位置，导致进入 呼叫 界面：



最后无奈，只能绕过这个bug，换用别的方式去点击元素：

用wda的query去查找元素，通过元素点击本身

```

# parentOtherSoup = callSoup.parent
# if parentOtherSoup:
#     parentParentOtherSoup = parentOtherSoup.parent
#     if parentParentOtherSoup:
#         cancelSoup = parentParentOtherSoup.find(
#             "XCUIElementTypeButton",
#             attrs={"enabled":"true", "visible":"true", "name": "取消"})
#     )
#     if cancelSoup:
#         clickCenterPosition(curSession, cancelSoup.attrs)
#         foundAndProcessedPopup = True

# above click position not work for 取消 !!!
# change to find 取消 then click element

cancelButtonQuery = {"type":"XCUIElementTypeButton", "enabled":"true", "visible":"true", "name": "取消"}
foundAndClicked = findAndClickElement(curSession, cancelButtonQuery)
foundAndProcessedPopup = foundAndClicked

```

才可以：点击取消 让弹框消失。

详见：

【已解决】自动抓包iOS的app无忧筹：弹框呼叫拨打电话

【后记1】

又在：

【未解决】自动抓包iOS的app京东金融：弹框想给您发送通知允许

遇到同样的问题：

bs4中搜索到了 允许 按钮，去点击 通过点击允许按钮的中间坐标值，结果实际上却是点击了：另外一个按钮 不允许。。。

然后无奈，只能想办法用wda的query去查询元素 允许，再通过元素点击 估计就可以了。

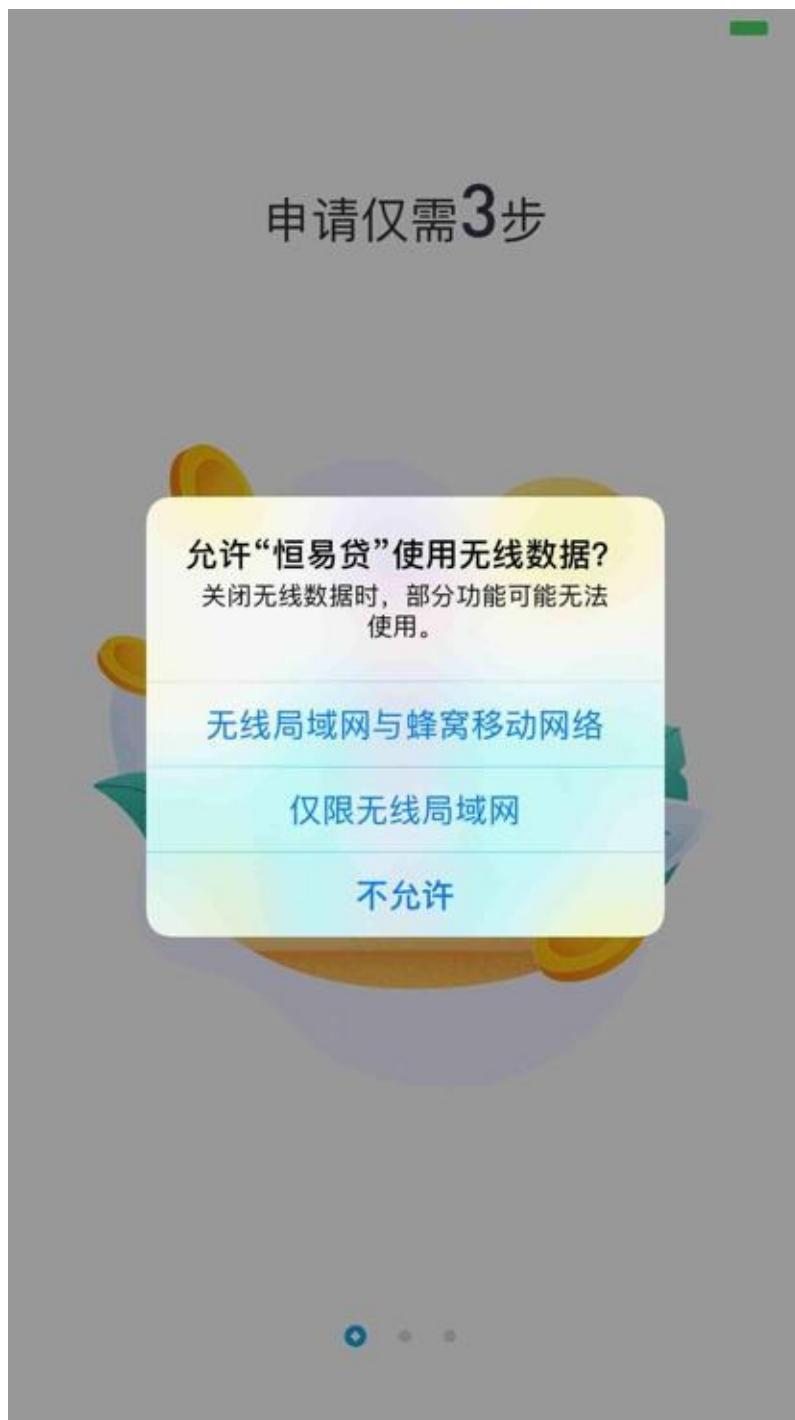
【后记2】

又在：

【未解决】自动抓包iOS的app恒易贷：弹框使用无线数据无线局域网与蜂窝移动网络

遇到同样问题：

对于页面：



都已经用代码：

```
wifiCellularChainList = [
    {
        "tag": "XCUIElementTypeAlert",
        "attrs": {"enabled":"true", "visible":"true"}
    },
    {
        "tag": "XCUIElementTypeOther",
        "attrs": {"enabled":"true", "visible":"true"}
    }
];
```

```

    {
        "tag": "XCUIElementTypeButton",
        "attrs": {"enabled":"true", "visible":"true", "name": "无线局域网与蜂窝移动网络"}
    },
]
wifiCellularSoup = utils.bsChainFind(soup, wifiCellularChainList)
if wifiCellularSoup:
    clickCenterPosition(curSession, wifiCellularSoup attrs)
    foundAndProcessedPopup = True
return foundAndProcessedPopup

```

查到并点击了 无线局域网与蜂窝移动网络 按钮的中间位置，但是实际上点击的是：不允许
导致后来app无法访问网络，再次启动app后，也提示请开启网络权限。

只能去改为， wda的元素查找，找到元素后，根据元素去click点击：

```

wifiCellularSoup = CommonUtils.bsChainFind(soup, wifiCellularChainList)
if wifiCellularSoup:
    # self.clickElementCenterPosition(wifiCellularSoup)
    # foundAndProcessedPopup = True

    # found 无线局域网与蜂窝移动网络 but actually click center position will click 不
    允许 !!!
    # change to wda query element then click by element
    curName = wifiCellularSoup.attrs["name"] # 好
    wifiCellularButtonQuery = {"type":"XCUIElementTypeButton", "enabled":"true", "name": curName}
    foundAndClicked = self.findAndClickElement(wifiCellularButtonQuery, isShowErrMsg=False)
    foundAndProcessedPopup = foundAndClicked
return foundAndProcessedPopup

```

才可以。

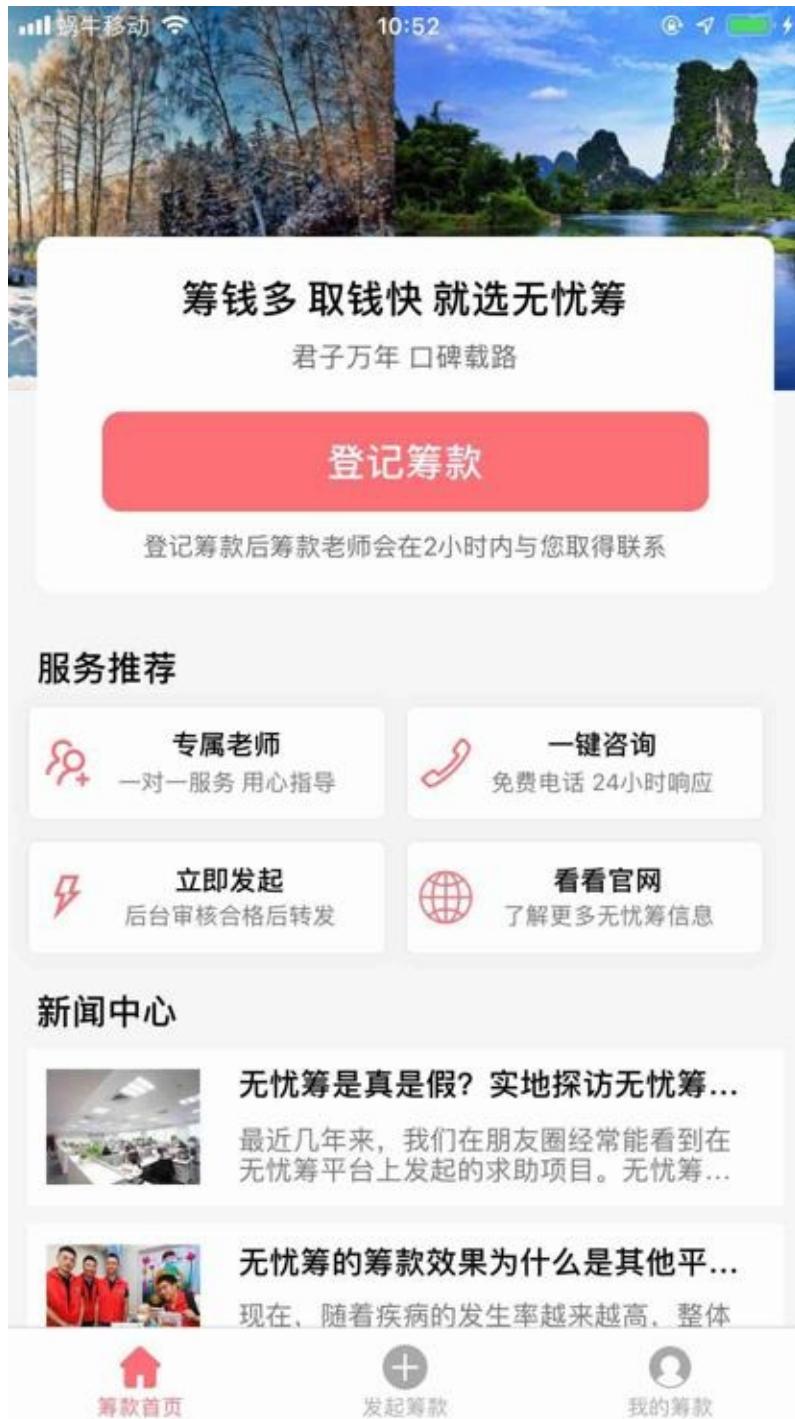
【后记3】

由于经常遇到此问题，所以来专门去提取逻辑到独立函数中，详见：

[元素处理 · iOS自动化测试利器：facebook-wda](#)

**坑：元素查找条件 都写的最完整，不能再详细了，但是却会出现
可以查询到 找到 多个元素**

比如页面：



左下角的 3个tab页的父级元素，对应 locator，调试出现警告：

```
[200515 14:23:27][ParsePage.py 1019] Found 2 same node from tag XCUIElementTypeOther, attr{'name': '筹款首页', tab, 1 of 3 发起筹款, tab, 2 of 3 我的筹款, tab, 3 of 3', 'label': '筹款首页, tab, 1 of 3 发起筹款, tab, 2 of 3 我的筹款, tab, 3 of 3', 'enabled': 'true', 'visible': 'true', 'x': '0', 'y': '687', 'width': '414', 'height': '49', 'type': 'XCUIElementTypeOther'}
```

提示上述locator可以找到2个元素，然后去xml源码中看看，果然是的：

```

<XCUIElementTypeOther type="XCUIElementTypeOther" name="筹款首页, tab, 1 of 3 发起筹款,
tab, 2 of 3 我的筹款, tab, 3 of 3" label="筹款首页, tab, 1 of 3 发起筹款, tab, 2 of 3 我的筹
款, tab, 3 of 3" enabled="true" visible="true" x="0" y="687" width="414" height="49">
    <XCUIElementTypeOther type="XCUIElementTypeOther" name="筹款首页, tab, 1 of 3 发起筹款,
tab, 2 of 3 我的筹款, tab, 3 of 3" label="筹款首页, tab, 1 of 3 发起筹款, tab, 2 of 3 我的筹
款, tab, 3 of 3" enabled="true" visible="true" x="0" y="687" width="414" height="49">
        ...

```

```

194
195   height="165"/>
    <XCUIElementTypeOther type="XCUIElementTypeOther".
    enabled="true" visible="false" x="207" y="1125" width="165".
    height="165"/>
196   </XCUIElementTypeOther>
197   </XCUIElementTypeOther>
198   </XCUIElementTypeOther>
199   </XCUIElementTypeScrollView>
200   </XCUIElementTypeOther>
201   </XCUIElementTypeOther>
202   </XCUIElementTypeOther>
203   </XCUIElementTypeOther>
204   </XCUIElementTypeOther>
205   </XCUIElementTypeOther type="XCUIElementTypeOther" name="筹款首页, tab, 1 of 3 发起
筹款, tab, 2 of 3 我的筹款, tab, 3 of 3" label="筹款首页, tab, 1 of 3 发起筹款,
tab, 2 of 3 我的筹款, tab, 3 of 3" enabled="true" visible="true" x="0" y="687"
width="414" height="49">
206     <XCUIElementTypeOther type="XCUIElementTypeOther" name="筹款首页, tab, 1 of 3
发起筹款, tab, 2 of 3 我的筹款, tab, 3 of 3" label="筹款首页, tab, 1 of 3 发起筹
款, tab, 2 of 3 我的筹款, tab, 3 of 3" enabled="true" visible="true" x="0".
y="687" width="414" height="49">
207       <XCUIElementTypeButton type="XCUIElementTypeButton" value="1" name="筹款首
页, tab, 1 of 3" label="筹款首页, tab, 1 of 3" enabled="true" visible="true"
x="0" y="689" width="138" height="48"/>
208       <XCUIElementTypeButton type="XCUIElementTypeButton" name="发起筹款, tab, 2
of 3" label="发起筹款, tab, 2 of 3" enabled="true" visible="true" x="138".
y="689" width="138" height="48"/>
209       <XCUIElementTypeButton type="XCUIElementTypeButton" name="我的筹款, tab, 3
of 3" label="我的筹款, tab, 3 of 3" enabled="true" visible="true" x="276".
y="689" width="138" height="48"/>

```

就是：底部3个按钮主菜单 的parent 和 parent的parent

-》 坑就是：

如果通过上述（最详尽的）条件去定位元素，则理论上是会出现多个的

-》 无法完美精准定位查询到某个想要的元素。

详见：

【未解决】自动抓包iOS的app：无忧筹点击首页的筹款首页后无法返回

wda获取到了switch的值，但是是错的

对于页面：



对应xml

```

<XCUIElementTypeCell type="XCUIElementTypeCell" value="0" enabled="true" visible="true" x="0" y="359" width="414" height="46">
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="404" width="414" height="1"/>
    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="鉴定" name="鉴定" label="鉴定" enabled="true" visible="true" x="20" y="359" width="323" height="46"/>
    <XCUIElementTypeSwitch type="XCUIElementTypeSwitch" value="0" name="鉴定" label="鉴定" enabled="true" visible="true" x="343" y="366" width="51" height="32"/>
</XCUIElementTypeCell>

```



已经可以用代码：

```
newAuthenticateValue = newManualProxyValue["authenticate"]
authSwitchQuery = {"type": "XCUIElementTypeSwitch", "name": "鉴定", "enabled": "true"}
authSwitchQuery["parent_class_chains"] = [parentCellClassChain]
foundAuth, respInfo = self.findElement(authSwitchQuery, timeout=0.1)
```

找到 鉴定 对应的switch

但是获取其value值：

```
authSwitchElement = respInfo
curAuthValue = authSwitchElement.value # '0'
```

竟然是： '0'

而不是真正实际的值： '1'

规避办法：

最后无奈只能改用别的方式（bs的find，获取到xml源码）去获取值

虽然速度慢点，但是至少值是准的：

```
curAuthValueStr = ""
# curAuthValue = authSwitchElement.value # '0'
# curAuthValueStr = str(curAuthValue)
# Special: sometime wda element value is WRONG, actual is '1', but got '0'
# so change to bs find then get value from page source xml
curPageXml = self.get_page_source()
soup = CommonUtils.xmlToSoup(curPageXml)
authSwitchChainList = [
    {
        "tag": "XCUIElementTypeTable",
        "attrs": self.FullScreenAttrDict
    },
    {
        "tag": "XCUIElementTypeCell",
        "attrs": {"enabled": "true", "visible": "true", "x": "0", "width": "%s" % self.X}
    },
    {
        "tag": "XCUIElementTypeSwitch",
        "attrs": {"enabled": "true", "visible": "true", "name": "鉴定"}
    }
]
authSwitchSoup = CommonUtils.bsChainFind(soup, authSwitchChainList)
if authSwitchSoup:
    curAuthValue = authSwitchSoup.attrs.get("value", None)
    if curAuthValue:
        curAuthValueStr = str(curAuthValue)
```

详见：

【已解决】facebook-wda获取鉴定的value值是错误的

wda找到元素，但是无法用clear_text清除值value值

界面中：



用代码已经找到 服务器 元素了：

```
newServerValue = newManualProxyValue["server"]
```

```

serverFieldQuery = {"type": "XCUIElementTypeTextField", "name": "服务器", "enabled": "true"}
serverFieldQuery["parent_class_chains"] = [ parentCellClassChain ]
isFound, respInfo = self.findElement(query=serverFieldQuery)
logging.debug("isFound=%s, respInfo=%s", isFound, respInfo)
if isFound:
    curElement = respInfo

```

但是去清除当前的值

```
curElement.clear_text()
```

却不起效果

最后无奈用 `set_text()` 传入多个 `\b`，通过一个个删除字符的方式实现了删除输入的值的效果

```

def iOSClearText(self, curElement):
    """iOS clear current element's text value
    Note: clear_text not working, so need use other workaround to do clear text

    Args:
        curElement (Element): wda element
    Returns:
        Raises:
    """
    # curElement.click()
    # curElement.clear_text()
    # curElement.tap_hold(2.0) # then try select All -> Delete
    backspaceChar = '\b'
    maxDeleteNum = 50
    curElement.set_text(maxDeleteNum * backspaceChar)
    return

```

调用：

```

curElement = respInfo
if isNeedClear:
    # before set new value, clear current value
    self.iOSClearText(curElement)
curElement.set_text(text)

```

间接实现clear text的效果。

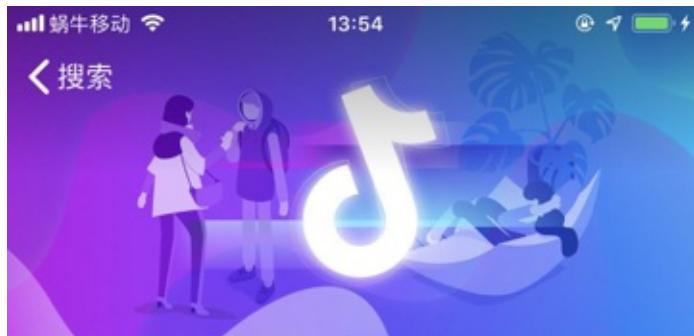
详见：

【已解决】facebook-wda中元素clear清除文本值无效

无法获取元素value值

类似于页面：

AppStore中，正在下载app



xml源码是：

```
<XCUIElementTypeButton type="XCUIElementTypeButton" value="18%" name="正在下载" label="正在  
下载" enabled="true" visible="true" x="154" y="308" width="74" height="30"/>
```

但是通过

文件： /Users/limao/.pyenv/versions/3.8.0/Python.framework/Versions/3.8/lib/python3.8/site-packages/wda/__init__.py

```
@property  
def value(self):  
    # curValue = self._prop('attribute/value')  
    curValue = self._prop('attribute/wdValue')
```

```

if DEBUG:
    print("curValue=%s" % curValue)
return curValue

```

对应着代码：

文件： refer/WebDriverAgent/WebDriverAgentLib/Commands/FBElementCommands.m

```

[[FBRoute GET:@"/element/:uuid/attribute/:name"] respondWithTarget:self action:@select
or(handleGetAttribute:)],

+ (id<FBResponsePayload>)handleGetAttribute:(FBRouteRequest *)request
{
    FBELEMENTCache *elementCache = request.session.elementCache;
    XCUIElement *element = [elementCache elementForUUID:request.parameters[@"uuid"]];
    if (nil == element) {
        return FBResponseWithStatus([FBCommandStatus staleElementReferenceErrorWithMessage:nil
                                         traceback:nil]);
    }
    id attributeValue = [element fb_valueForWDAttributeName:request.parameters[@"name"]];
    attributeValue = attributeValue ?: [NSNull null];
    return FBResponseWithObject(attributeValue);
}

```

却获取不到value值，始终是null：

```

20200609 01:49:34 connectionpool.py:428 DEBUG http://localhost:8100 "POST /session/710DB6C7-3669-4677-B479-C006692CC3F6/element HTTP/1.1" 200 587
20200609 01:49:34 __init__.py:178 DEBUG Return (213ms):
{
    "value" : {
        "element-6066-11e4-a52e-4f735466cecf" : "32000000-0000-0000-4122-000000000000",
        "attribute\\visible" : true,
        "attribute\\name" : "正在下载",
        "attribute\\value" : null,
        "attribute\\accessible" : true,
        "text" : "正在下载",
        "label" : "正在下载",
        "rect" : {
            "y" : 308,
            "x" : 154,
            "width" : 74,
            "height" : 30
        },
        "type" : "XCUIElementTypeButton",
        "name" : "XCUIElementTypeButton",
        "ELEMENT" : "32000000-0000-0000-4122-000000000000"
    },
    "sessionId" : "710DB6C7-3669-4677-B479-C006692CC3F6"
}

```

```
20200609 01:49:48 connectionpool.py:428 DEBUG http://localhost:8100 "GET /session/710DB6C7-3669-4677-B479-C006692CC3F6/element/32000000-0000-0000-4122-000000000000/attribute/wdValue HTTP/1.1" 200 76
20200609 01:49:48 __init__.py:178 DEBUG Return (204ms): {
    "value": null,
    "sessionId": "710DB6C7-3669-4677-B479-C006692CC3F6"
}
```

详见：

【未解决】研究facebook-wda和WebDriverAgent中attribute/value始终是null无法获取有效值

偶尔页面中有内容刷新，动画进行中，则无法方便的获取到页面源码

详见：

【未解决】WebDriverAgent获取iPhone页面源码报错：Code 5 Error kAXErrorIPCTimeout getting snapshot for element

页面源码中，个别元素的最大x值超出屏幕

比如页面：



中的店铺热卖

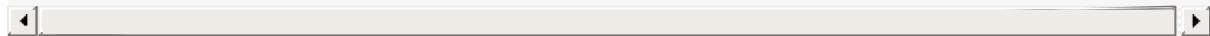
相关部分xml是：

```

<XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="44" y="617"
" width="111" height="50">
    <XCUIElementTypeButton type="XCUIElementTypeButton" name="牛尔天猫" label="牛尔天猫" ena
bled="true" visible="true" x="44" y="617" width="111" height="49"/>
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="59"
y="666" width="96" height="1"/>
</XCUIElementTypeCell>
<XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="154" y="6
17" width="111" height="50">
    <XCUIElementTypeButton type="XCUIElementTypeButton" name="会员福利" label="会员福利" ena
bled="true" visible="true" x="154" y="617" width="111" height="49"/>
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="169"
y="666" width="96" height="1"/>
</XCUIElementTypeCell>

```

```
y="666" width="96" height="1"/>
</XCUIElementTypeCell>
<XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="264" y="6
17" width="112" height="50">
    <XCUIElementTypeButton type="XCUIElementTypeButton" name="店铺热卖" label="店铺热卖" ena
bled="true" visible="true" x="264" y="617" width="112" height="49"/>
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="279"
y="666" width="97" height="1"/>
</XCUIElementTypeCell>
```



店铺热卖 的: $x1=x0+width=264+112=376$ 大于 屏幕宽度375

导致原先代码逻辑判断出错：以为元素不在bottom底部区域，而被过滤掉，找不到菜单

最后是额外加了特殊处理，才可以保留和找到此菜单

详见：

【已解决】自动抓包iOS公众号：niuer-tmall中丢失部分主菜单

元素query时不完全支持visible属性

最新的结果也支持：有时候支持，有时候不支持

比如对于页面：



中，xml是：

```
<XCUIElementTypeButton type="XCUIElementTypeButton" name="牛尔天猫" label="牛尔天猫" enabled="true" visible="true" x="44" y="617" width="111" height="49"/>
```

query查询条件中，加了visible：

```
{'enabled': 'true', 'height': '49', 'label': '牛尔天猫', 'name': '牛尔天猫', 'type': 'XCUIElementTypeButton', 'width': '111', 'x': '44', 'y': '617', 'visible': 'true'}
```

就找不到，log是：

```
[200430 17:00:39] [__init__.py 164] Shell: curl -X POST -d '{"using": "class chain", "value": "
```

```
": "**/XCUIElementTypeButton[`name == '\u725b\u5c14\u5929\u732b' AND label == '\u725b\u5c14\u5929\u732b' AND enabled == 1 AND visible == 1 AND rect.x == 44 AND rect.y == 617 AND rect.width == 111 AND rect.height == 49`]" }' 'http://localhost:8100/session/E91B12B7-E999-4945-BD4F-57E818C91EA4/element'
[200430 17:00:39][connectionpool.py 221] Starting new HTTP connection (1): localhost:8100
[200430 17:00:40][connectionpool.py 428] http://localhost:8100 "POST /session/E91B12B7-E999-4945-BD4F-57E818C91EA4/element HTTP/1.1" 404 6971
[200430 17:00:40][__init__.py 178] Return (175ms): {
    "value" : {
        "error" : "no such element",
        "message" : "unable to find an element using 'class chain', value '**/XCUIElementTypeButton[`name == '牛尔天猫' AND label == '牛尔天猫' AND enabled == 1 AND visible == 1 AND rect.x == 44 AND rect.y == 617 AND rect.width == 111 AND rect.height == 49`]" ,
        ...
    }
}
```

去掉visible:

```
{'enabled': 'true', 'height': '49', 'label': '牛尔天猫', 'name': '牛尔天猫', 'type': 'XCUIElementTypeButton', 'width': '111', 'x': '44', 'y': '617'}
```

就能找到。

详见：

[【已解决】Python中facebook-wda和WebDriverAgent中是否可以支持displayed以及是否能替换visible](#)

[【已解决】自动抓包iOS公众号：niuer-tmall定位主菜单失败](#)

[【已解决】合并最新版WebDriverAgent后测试是否支持元素的visible属性的query查询](#)

偶尔代码无法运行，要看看服务端test manager是否正常

调试时发现偶尔卡死：

```
x limao@limao ~ /dev /crawler/appAutoCrawler/AppCrawler/iOSAutomation master • env PTVS_LAUNCHER_PORT=50892 /User
s/limao/.pyenv/versions/3.8.0/Python.framework/Versions/3.8/bin/python /Users/limao/.vscode/extensions/ms-python.python-2020.2.64397/p
thonFiles/lib/python/new_ptvsd/no_wheels/ptvsd/launcher /Users/limao/dev/FiboDT/crawler/appAutoCrawler/AppCrawler/iOSAutomation/wda
st/wdaTest.py
gServerUrl=http://192.168.31.43:8100
gwdaClient=<wda.Client object at 0x10bc41b20>
Shell: curl -X POST -d '{"desiredCapabilities": {"bundleId": "com.tencent.xin", "shouldWaitForQuiescence": false}, "capabilities": {"a
lwaysMatch": {"bundleId": "com.tencent.xin", "shouldWaitForQuiescence": false}}}' 'http://192.168.31.43:8100/session'
```

始终无法继续运行了。

以为代码改动出了问题。

后来发现是：服务端挂了：

```
Testing failed:
WebDriverAgentRunner:
    testRunner encountered an error (Encountered a problem with the test runner after
launch. (Underlying error: Lost connection to DTServiceHub))
** TEST FAILED **
```

```
** If you believe this error represents a bug, please attach the result bundle at /Users/Limao/Library/Developer/Xcode/DerivedData/WebDriverAgent-doxykunuiuxsdzeisbkcuadwxyab/Logs/Test/WebDriverAgentRunner-2020.03.13_16-31-13+0800.xcresult

2020-03-13 17:16:35.851 xcodebuild[3621:41314] [MT] IDETestOperationsObserverDebug: 2720.400 elapsed -- Testing started completed.
2020-03-13 17:16:35.851 xcodebuild[3621:41314] [MT] IDETestOperationsObserverDebug: 0.000 sec, +0.000 sec -- start
2020-03-13 17:16:35.852 xcodebuild[3621:41314] [MT] IDETestOperationsObserverDebug: 2720.400 sec, +2720.400 sec -- end
2020-03-13 17:16:35.853 xcodebuild[3621:41314] Error Domain=com.apple.platform.iphoneos Code=-13 "Lost connection to DTServiceHub" UserInfo={NSLocalizedDescription=Lost connection to DTServiceHub}

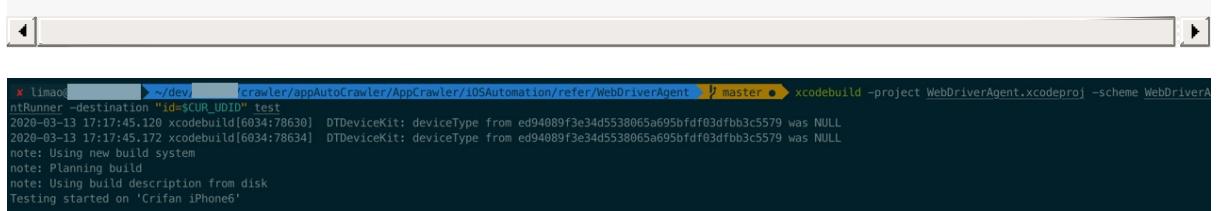
Test session results, code coverage, and logs:
  /Users/limao/Library/Developer/Xcode/DerivedData/WebDriverAgent-doxykunuiuxsdzeisbkcuadwxyab/Logs/Test/Test-WebDriverAgentRunner-2020.03.13_16-31-13+0800.xcresult

Testing failed:
  WebDriverAgentRunner:
    testRunner encountered an error (Encountered a problem with the test runner after launch. (Underlying error: Lost connection to DTServiceHub))

** TEST FAILED **
```

所以去重启服务：

```
xcodebuild -project WebDriverAgent.xcodeproj -scheme WebDriverAgentRunner -destination "id=$CUR_UDID" test
```



```
* limao:~/dev/crawler/appAutoCrawler/AppCrawler/iOSAutomation/refer(WebDriverAgentRunner) master ➜ xcodebuild -project WebDriverAgent.xcodeproj -scheme WebDriverAgentRunner -destination "id=$CUR_UDID" test
2020-03-13 17:17:45.120 xcodebuild[6034:78630] DTDeviceKit: deviceType from ed94089f3e34d5538065a695bffd03dfbb3c5579 was NULL
2020-03-13 17:17:45.172 xcodebuild[6034:78634] DTDeviceKit: deviceType from ed94089f3e34d5538065a695bffd03dfbb3c5579 was NULL
note: Using new build system
note: Planning build
note: Using build description from disk
Testing started on 'Crifan iPhone6'
```

直到看到：

```
2020-03-13 17:17:57.615841+0800 WebDriverAgentRunner-Runner[725:273028] ServerURLHere- http://192.168.31.43:8100 -ServerURLHere
2020-03-13 17:17:57.659059+0800 WebDriverAgentRunner-Runner[725:273201] Using singleton test manager
```

即可。

crifan.com，使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2020-08-09 10:13:38

获取源码xml

iOS中最大的坑，就是获取页面源码 xml 期间，遇到的各种问题。

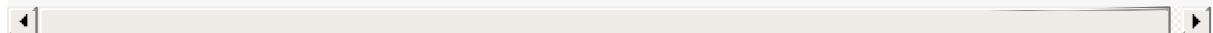
坑：即使查询条件和xml中内容正确匹配，也查询不到

对于页面：



xml是：

```
<XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="可申请(元) 200,000" name="可申请(元) 200,000" label="可申请(元) 200,000" enabled="true" visible="true" x="85" y="226" width="244" height="102"/>
```



去用：

```
{'value': '可申请(元) 200,000', 'name': '可申请(元) 200,000', 'label': '可申请(元) 200,000', 'enabled': 'true', 'x': '85', 'y': '226', 'width': '244', 'height': '102', 'type': 'XCUIElementTypeStaticText'}
```

以及 去掉y的：

```
{'value': '可申请(元) 200,000', 'name': '可申请(元) 200,000', 'label': '可申请(元) 200,000', 'enabled': 'true', 'x': '85', 'width': '244', 'height': '102', 'type': 'XCUIElementTypeStaticText'}
```

都查不到元素。

不过，去掉value, name, label后：

```
{'enabled': 'true', 'x': '85', 'y': '226', 'width': '244', 'height': '102', 'type': 'XCUIElementTypeStaticText'}
```

是可以查询到元素的，所以很是诡异。

其原因，自己推测是此处的 (value等) 值有问题

但是具体的值是不是我猜测的

可申请(元) 200000

则无需，也懒得再去试了。

更重要的是，对于：

可申请(元) 200,000

页面上的内容的显示，是肉眼可见的分成了2部分

可申请(元)

200,000

且显示的样式都不同

-> 所以十分怀疑是：

iOS内部的元素和代码，其实本身就是这2部分是分开的

只不过是输出xml时，混在了一起

-> 导致通过value (name, label) 才找不到元素的

-> 去掉value等值后，只用x、y等坐标值，就能找到：说明是对应着页面上的其中某一个元素
要么是 可申请(元)，或者是200,000

总之是：

iOS内部页面内容，和输出xml代码之间，一直做的很垃圾。

或者说故意做的很垃圾，让你很难自动化测试iOS。

详见：

【不去解决】自动抓包iOS的app恒易贷：找不到元素可申请元200000

坑：界面上按钮有文字，但是源码中没有文字

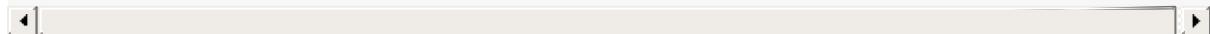
界面上：



本来希望去：写规则去查找button，且name是立即进入

结果源码中

```
<XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="736">
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="736">
        <XCUIElementTypeScrollView type="XCUIElementTypeScrollView" enabled="true" visible="true" x="0" y="0" width="414" height="736">
            <XCUIElementTypeImage type="XCUIElementTypeImage" name="guide2" enabled="true" visible="true" x="0" y="0" width="414" height="736"/>
        </XCUIElementTypeScrollView>
        <XCUIElementTypeButton type="XCUIElementTypeButton" name="loading btn" label="loading btn" enabled="true" visible="true" x="141" y="628" width="132" height="30"/>
    </XCUIElementTypeOther>
</XCUIElementTypeOther>
```



没有我们希望的文字：立即进入

注：目测看起来，这个 立即进入 的button的文字 不是属于button图片本身，而是普通文字，只不过 xml源码中，的确找不到

这样就影响了后续代码逻辑的判断，无法准确判断当前页面的按钮，是否是最后一页了。

详见：

【未解决】自动抓包iOS的app：左滑引导页进入首页

不爽的点：页面类似，但xml源码差异很大

对于页面：



但是对应xml：

```
<XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="64"
" width="414" height="623">
    <XCUIElementTypeTable type="XCUIElementTypeTable" name="空列表" label="空列表" enabled="true"
visible="false" x="0" y="64" width="414" height="623"/>
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="64"
width="414" height="623">
        <XCUIElementTypeImage type="XCUIElementTypeImage" name="com_network_err" enabled="true"
visible="false" x="152" y="164" width="110" height="110"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="网络不稳定" name="网络不稳定"
label="网络不稳定" enabled="true" visible="true" x="0" y="287" width="414" height="17"/>
        <XCUIElementTypeButton type="XCUIElementTypeButton" name="刷新试试" label="刷新试试"
enabled="true" visible="true" x="147" y="323" width="120" height="41"/>
    </XCUIElementTypeOther>
</XCUIElementTypeOther>
```



很明显，页面中的 刷新试试 明显是一个按钮，是没问题的

-> 后续就容易写规则去匹配和处理

但是后来遇到和上面很类似的页面：



可见页面上 再刷新下 也是一个按钮

但发现xml却是：

```
<XCUIElementTypeOther type="XCUIElementTypeOther" name="系统正在开小差，请稍后再试 再刷新下" 1
label="系统正在开小差，请稍后再试 再刷新下" enabled="true" visible="true" x="0" y="64" width="4
14" height="672">
```

```
<XCUIElementTypeOther type="XCUIElementTypeOther" name="系统正在开小差, 请稍后再试 再刷新下" label="系统正在开小差, 请稍后再试 再刷新下" enabled="true" visible="true" x="0" y="64" width="414" height="672">
    <XCUIElementTypeOther type="XCUIElementTypeOther" name="系统正在开小差, 请稍后再试" label="系统正在开小差, 请稍后再试" enabled="true" visible="true" x="114" y="240" width="186" height="136">
        <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="152" y="240" width="110" height="111"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="系统正在开小差, 请稍后再试" name="系统正在开小差, 请稍后再试" label="系统正在开小差, 请稍后再试" enabled="true" visible="true" x="114" y="360" width="186" height="16"/>
    </XCUIElementTypeOther>
    <XCUIElementTypeOther type="XCUIElementTypeOther" name="再刷新下" label="再刷新下" enabled="true" visible="true" x="147" y="375" width="120" height="61">
        <XCUIElementTypeOther type="XCUIElementTypeOther" name="再刷新下" label="再刷新下" enabled="true" visible="true" x="147" y="395" width="120" height="41"/>
    </XCUIElementTypeOther>
</XCUIElementTypeOther>
</XCUIElementTypeOther>
```

再刷新下却是一个 XCUIElementTypeOther，而不是XCUIElementTypeButton

-> 后续代码去处理和写匹配逻辑，就显得很不顺，让人很不爽。

-> 如果也是和前面一样的XCUIElementTypeButton，就容易统一成一个逻辑去处理，更加通用，效率更高。

-> 现在没法统一，效率很低，逻辑上显得很冗余

总体结论：

页面上的元素，和xml源码内容，很多时候，对不上，甚至完全对不上，驴唇不对马嘴的感觉。

详见：

【未解决】自动抓包iOS的app京东金融：网络不稳定刷新试试

【未解决】自动抓包iOS的app京东金融：系统正在开小差再刷新下

坑：有些页面 获取到的源码实际上是空的 没有包含页面元素的源码

比如页面：

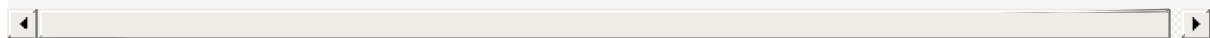


希望获取源码中包含弹框部分的内容

但是实际上获取到的是：

```
<?xml version="1.0" encoding="UTF-8"?>
<XCUIElementTypeApplication type="XCUIElementTypeApplication" name="益路同行" label="益路同行"
```

```
" enabled="true" visible="true" x="0" y="0" width="414" height="736">
    <XCUIElementTypeWindow type="XCUIElementTypeWindow" enabled="true" visible="true" x="0" y="0" width="414" height="736">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="736">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="736">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="736">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="736"/>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    </XCUIElementTypeWindow>
    <XCUIElementTypeWindow type="XCUIElementTypeWindow" enabled="true" visible="false" x="0" y="0" width="414" height="736">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="0" width="414" height="736"/>
    </XCUIElementTypeWindow>
    <XCUIElementTypeWindow type="XCUIElementTypeWindow" enabled="true" visible="false" x="0" y="0" width="0" height="0"/>
    <XCUIElementTypeWindow type="XCUIElementTypeWindow" enabled="true" visible="false" x="0" y="0" width="414" height="736">
        <XCUIElementTypeStatusBar type="XCUIElementTypeStatusBar" enabled="true" visible="false" x="0" y="0" width="414" height="20"/>
    </XCUIElementTypeWindow>
    <XCUIElementTypeWindow type="XCUIElementTypeWindow" enabled="true" visible="false" x="0" y="0" width="414" height="736">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="0" width="414" height="736"/>
    </XCUIElementTypeWindow>
</XCUIElementTypeApplication>
```



即：

中间主体内容是空的

没有包含我们希望看到的 弹框部分

详见：

【未解决】自动抓包iOS的app益路同行：弹框退出游戏

坑：页面中图片明显可见，但是xml源码中visible=false表示不可见

页面中的中间部分的2个图片：



此处xml源码竟然是：

```

<XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="0" y="280" width="414" height="320">
    <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="false" x="15" y="290" width="384" height="181"/>
    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="【善友筹】（已审核）父母懂事的孩子，成绩优异的班长，因白血病不得不离开学校和小伙伴，大家一起帮帮他渡过难关，定会滴水之恩，涌泉相报！" name="【善友筹】（已审核）父母懂事的孩子，成绩优异的班长，因白血病不得不离开学校和小伙伴，大家一起帮帮他渡过难关，定会滴水之恩，涌泉相报！" label="【善友筹】（已审核）父母懂事的孩子，成绩优异的班长，因白血病不得不离开学校和小伙伴，大家一起帮帮他渡过难关，定会滴水之恩，涌泉相报！" enabled="true" visible="true" x="15" y="475" width="384" height="60"/>
    <XCUIElementTypeProgressIndicator type="XCUIElementTypeProgressIndicator" value="8%" name="进度" label="进度" enabled="true" visible="true" x="15" y="534" width="334" height="23"/>
    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="发起人:邱嘉棋" name="发起人:邱嘉棋" label="发起人:邱嘉棋" enabled="true" visible="true" x="15" y="559" width="117" height="21"/>
    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="认筹金额:1693" name="认筹金额:1693" label="认筹金额:1693" enabled="true" visible="true" x="117" y="559" width="150" height="21"/>
    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="剩余:23天" name="剩余:23天" label="剩余:23天" enabled="true" visible="true" x="267" y="559" width="117" height="21"/>
    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="8%" name="8%" label="8%" enabled="true" visible="true" x="334" y="534" width="100" height="21"/>
</XCUIElementTypeCell>
```

其中

```
<XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="false" x="15" y="290" width="384" height="181"/>
```

即：

只有一个Image节点，（当前可能本身就是一张图，但是从app中看起来不像，还是像2张图）并且还是visible=false，即不可见！

你妹的，那还怎么解析出有效节点，根本没法提取有效节点，和后续抓取。

详见：

【未解决】自动抓包工具抓包iOS的app：善友筹

坑：app内部某一层的页面中的xml源码，竟然还保留（之前的几层）父级的元素

比如

某个二级页面：

康爱公社-二级页面-百万医保补充互助社.jpg



其中，正常的符合预期的是，页面xml源码中，有页面中的元素，比如

顶部的第二排的 互助公约 资助公示 本期分摊 联系客服 等

但是点击了 资助公示 后，进入 三级页面：

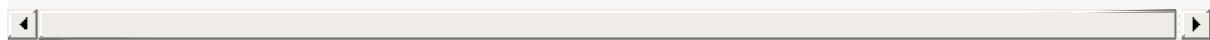
康爱公社-三级页面-资助公示-弹框提醒.jpg



竟然其中xml源码中，还有 前一页的页面元素：



```
<XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="218" y="280" width="67" height="45">
    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="本期分摊" name="本期分摊" label="本期分摊" enabled="true" visible="true" x="218" y="300" width="67" height="21"/>
</XCUIElementTypeOther>
```



其中可见，不仅存在之前页面的元素的xml，且竟然是visible=true，即：

表示当前页面可见。但是实际上不可见，不可能看到，前面几级页面的内容。

-》导致后续的基于xml源码判断元素的逻辑，就不可用了。完全混乱了。

即：在第三级页面，也能找到第二级，甚至第一级页面的元素，以为是在第二级或第一级页面呢，无需返回，即可找到并点击相关元素，而实际上页面上，并不是第二级或第一级页面，屏幕上并没有这些元素。

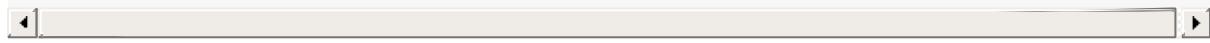
使得后续页面跳转，完全失效。无法继续正常逻辑。

仔细去看xml源码中发现，有个特点：

会存在 pages/xxx/xxx 之类的元素：

```
<XCUIElementTypeOther type="XCUIElementTypeOther" name="pages/mutual/introduce/introduce[4]" label="pages/mutual/introduce/introduce[4]" enabled="true" visible="false" x="0" y="64" width="414" height="3058">
    . . .

```



且不止一个：

```
<XCUIElementTypeOther type="XCUIElementTypeOther" name="pages/home/notice[5]" label="pages  
/home/notice[5]" enabled="true" visible="true" x="0" y="64" width="414" height="891">
```

.....

其中有几个 page/xxx

-> 存在 当前页面 实际上 包含了 几个（前后一共几级的） 页面的xml源码

详见：

【无法解决】iOS抓包app康爱公社：第三级页面中也能点击到第一级页面中的元素导致页面无法返回

【规避解决】iOS抓包app康爱公社：第三级别RestPage互助公约子页面无法返回

获取页面经常出现各种问题

比如：

【未解决】WebDriverAgent获取iPhone页面源码报错：Code 5 Error kAXErrorIPCTimeout getting snapshot for element

目前的结论是：

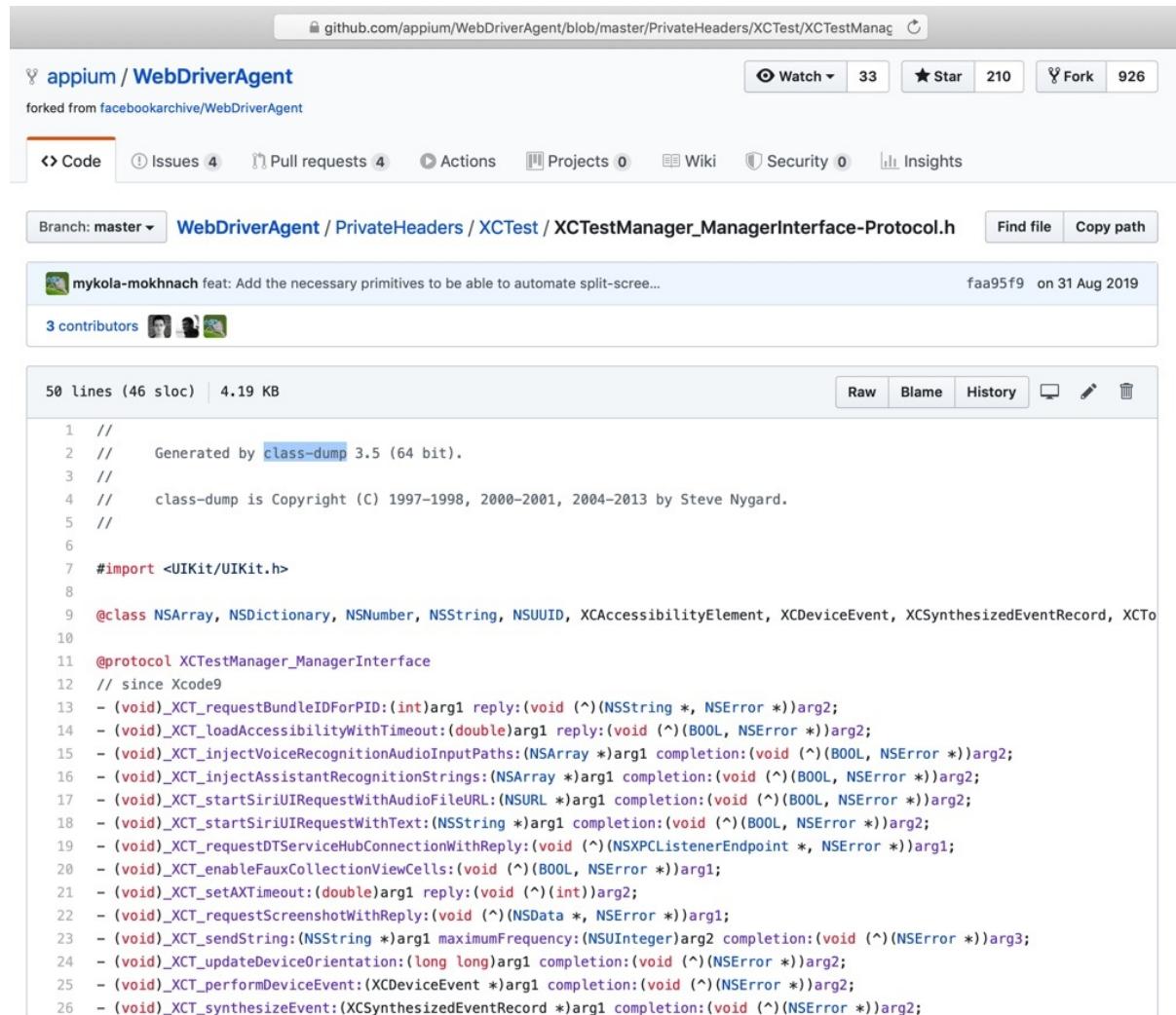
Apple方面，对于iOS设备的自动化测试，本身就是：从底层不太愿意支持

所以很多API接口，尤其是元素的是否可见的visible属性，就很难获取到

Appium的实现方式都是，想办法用第三方工具class-dump，RuntimeBrowser等从库中导出头文件，才看到有哪些API：

比如：

[WebDriverAgent/XCTestManager_ManagerInterface-Protocol.h at master · appium/WebDriverAgent](#)



appium / WebDriverAgent
forked from facebookarchive/WebDriverAgent

Watch 33 Star 210 Fork 926

Code Issues 4 Pull requests 4 Actions Projects 0 Wiki Security 0 Insights

Branch: master • WebDriverAgent / PrivateHeaders / XCTest / XCTestManager_ManagerInterface-Protocol.h Find file Copy path

mykola-mokhnach feat: Add the necessary primitives to be able to automate split-scre... faa95f9 on 31 Aug 2019

3 contributors

50 lines (46 sloc) | 4.19 KB Raw Blame History

```
1 //  
2 // Generated by class-dump 3.5 (64 bit).  
3 //  
4 // class-dump is Copyright (C) 1997-1998, 2000-2001, 2004-2013 by Steve Nygard.  
5 //  
6 //  
7 #import <UIKit/UIKit.h>  
8 //  
9 @class NSArray, NSDictionary, NSNumber, NSString, NSUUID, XCAccessibilityElement, XCDeviceEvent, XCSynthesizedEventRecord, XCTo  
10  
11 @protocol XCTestManager_ManagerInterface  
12 // since Xcode9  
13 - (void)_XCT_requestBundleIDForPID:(int)arg1 reply:(void (^)(NSString *, NSError *))arg2;  
14 - (void)_XCT_loadAccessibilityWithTimeout:(double)arg1 reply:(void (^)(BOOL, NSError *))arg2;  
15 - (void)_XCT_injectVoiceRecognitionAudioInputPaths:(NSArray *)arg1 completion:(void (^)(BOOL, NSError *))arg2;  
16 - (void)_XCT_injectAssistantRecognitionStrings:(NSArray *)arg1 completion:(void (^)(BOOL, NSError *))arg2;  
17 - (void)_XCT_startSiriUIRequestWithAudiofileURL:(NSURL *)arg1 completion:(void (^)(BOOL, NSError *))arg2;  
18 - (void)_XCT_startSiriUIRequestWithText:(NSString *)arg1 completion:(void (^)(BOOL, NSError *))arg2;  
19 - (void)_XCT_requestDTServiceHubConnectionWithReply:(void (^)(NSXPCLListenerEndpoint *, NSError *))arg1;  
20 - (void)_XCT_enableFauxCollectionViewCells:(void (^)(BOOL, NSError *))arg1;  
21 - (void)_XCT_setAXTimeout:(double)arg1 reply:(void (^)(int))arg2;  
22 - (void)_XCT_requestScreenshotWithReply:(void (^)(NSData *, NSError *))arg1;  
23 - (void)_XCT_sendString:(NSString *)arg1 maximumFrequency:(NSUInteger)arg2 completion:(void (^)(NSError *))arg3;  
24 - (void)_XCT_updateDeviceOrientation:(long long)arg1 completion:(void (^)(NSError *))arg2;  
25 - (void)_XCT_performDeviceEvent:(XCDeviceEvent *)arg1 completion:(void (^)(NSError *))arg2;  
26 - (void)_XCT_synthesizeEvent:(XCSynthesizedEventRecord *)arg1 completion:(void (^)(NSError *))arg2;
```

然后利用这些私有的API，去实现想要的功能。

所以往往是：兼容性很差

尤其是iOS 升级到了13后，兼容性极其差，尤其是snapshot，即获取页面源码方面，会出现各种各样的问题，报各种各样的错

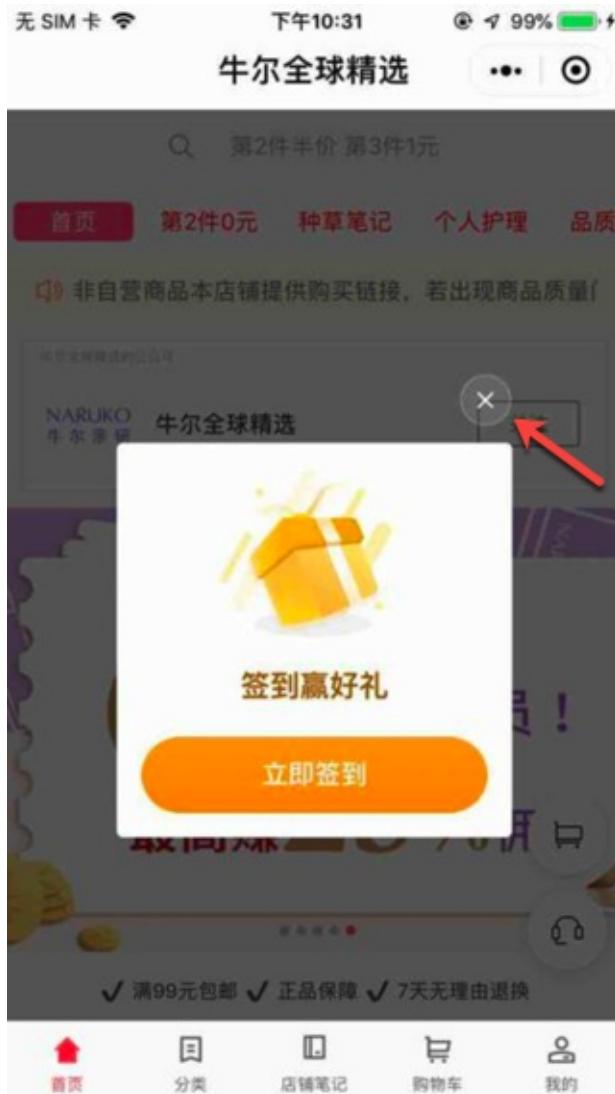
- Code 5 Error kAXErrorIPCTimeout getting snapshot for element
- Code=5 "Error -25216 getting snapshot for element
- Code=5 "Error kAXErrorServerNotFound getting snapshot for element

目前看来：

- 问题最多的：ios 13
- 稍微好一点的是：ios 12 或 ios 11

有些页面元素根本就无法获取到xml源码

比如：



相关部分源码是：

```
<XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="5" y="2591" width="365" height="33"/>
<XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="142" y="401" width="91" height="26">
    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="签到赢好礼" name="签到赢好礼" label="签到赢好礼" enabled="true" visible="true" x="142" y="400" width="91" height="28"/>
</XCUIElementTypeOther>
<XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="82" y="445" width="211" height="47">
    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="立即签到" name="立即签到" label="立即签到" enabled="true" visible="true" x="155" y="456" width="65" height="24"/>
</XCUIElementTypeOther>
```

即：只有 签到赢好礼 和 立即签到

根本就找不到 弹框右上角的 X关闭 按钮的xml源码

-》导致无法定位按钮元素，无法点击关闭弹框

详见：

【未解决】自动抓包iOS公众号：小程序中可关闭弹框签到赢好礼

无法获取完整页面源码

比如：



现象是：

- 微信公众号搜索页 搜 unesunes后：
 - iOS 11的iPhone6P：获取不到完整源码
 - iOS 13的iPhone8P：能获取到源码
 - iOS 12的iPhone6：能获取到源码

其中iOS 11的iPhone6P获取源码期间，test manager能看到错误log

```
t = 4793.27s Find: Identity Binding
2020-04-29 10:26:17.325280+0800 WebDriverAgentRunner-Runner[1017:316917] Cannot take the snapshot of Window after 65 seconds
2020-04-29 10:26:17.325547+0800 WebDriverAgentRunner-Runner[1017:316917] Internal error: Error Domain com.apple.dt.xctest.automation-support.error Code 5 "Error -25216 getting snapshot for element <AXUIElementRef 0x10ddae470> {pid=1148} {uid=[ID:9999 hash:0xf22fd4580000 0000]}" UserInfo {NSLocalizedDescription Error -25216 getting snapshot for element <AXUIEl
```

```
ementRef 0x10ddae470 <pid 1148> {uid [ID:9999 hash:0xf22fd45800000000]})  
2020-04-29 10:26:17.325677+0800 WebDriverAgentRunner-Runner[1017:316917] Skipping source dump for 'Window' because its snapshot cannot be resolved
```

按道理，应该是换iOS 13的iPhone8P或iOS 12的iPhone6，但是之前又都是由于有各种问题，才换这个iOS 11的iPhone6P的：

- iOS 13 的 iPhone8P：
 - 优理氏的某个小程序的客服聊天页：获取源码，不仅是失败，而是会导致test manager崩溃
 - 不仅是崩溃，重启后也无效
 - 需要重新卸载后重新安装WebDriverAgentRunner-Runner才行
 - 但是依旧是获取源码导致崩溃，陷入死循环
- iOS 12 的 iPhone6：
 - 获取某些页面速度很慢
 - 竟然比（本身相对比较卡顿，不流畅的）iOS 11的iPhone6，还慢
 - 很久之前，更新WebDriverAgent代码之前
 - 动卡空间，点击 关注（还是进入）公众号
 - 会导致微信崩溃
 - 最新：合并最新WebDriverAgent代码后，目前暂时不崩溃了
 - 但是还是获取很多页面速度比较慢

总之是：

现在虽然有3个iPhone，不同硬件尺寸，不同iOS版本，竟然没有一个顺利运行的，各有各的问题

最终：

暂时只能换iPhone而规避解决：

换 iOS 13的iPhone8P 或 iOS 12的iPhone6，都可以：获取到完全的页面源码

详见：

【规避解决】自动抓包iOS公众号：获取微信公众号unesunes搜索结果页面源码失败

获取页面源码：偶尔会导致微信崩溃

最早的iPhone 6 + iOS 12.4.5，获取 动卡空间关注页 点击 关注（还是进入）公众号后，结果微信崩溃

后来不崩溃了

可能的原因：更新了WebDriverAgent代码，内部解决或规避了之前的某个（WebDriverAgent或iOS或Xcode的API本身的）bug？

详见：

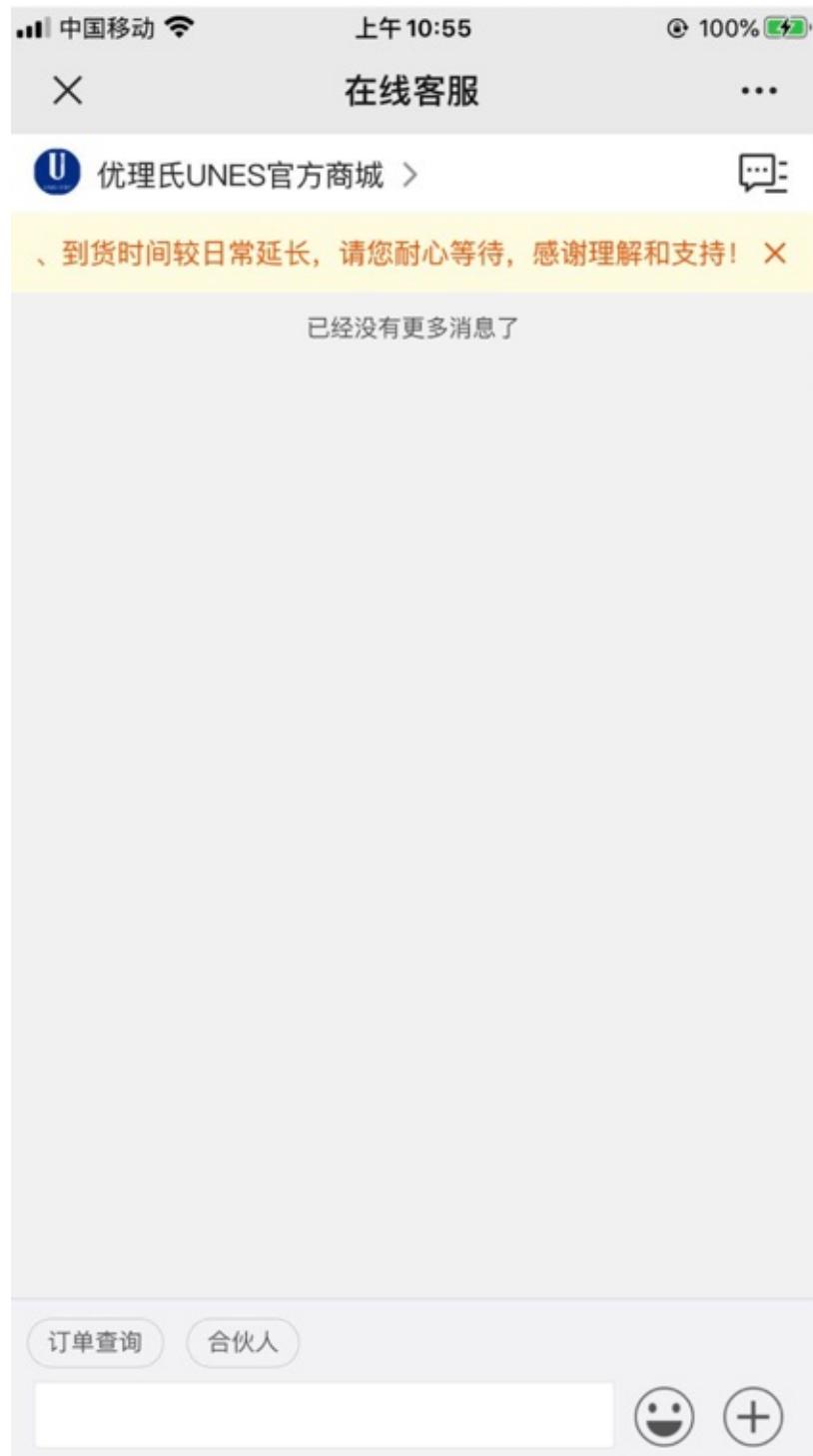
【记录】更新WebDriverAgent后测试iOS 12的iPhone6抓包微信公众号

获取页面源码：不仅无法获取源码，还会导致WebDriverAgent崩溃，要卸载后重新安装

WebDriverAgentRunner-Runner后才能重新使用

对于iOS 13.3.1的iPhone8P

去获取页面：



的源码时，会导致WebDriverAgent崩溃：

```

2020-04-28 10:49:13.283259+0800 WebDriverAgentRunner-Runner[3546:2307345] [xpc.exceptions]
NSXPConnection: 0x282540240 connection to service on pid 336 named com.apple.testmanagement
rd: Exception caught during decoding of reply to message '_XCT_fetchSnapshotForElement:att
ributes:parameters:reply:', dropping incoming message and calling failure block.

Ignored Exception: Exception while decoding argument 0 (#1 of invocation):
NSInvocation: 0x280fa4900
return value: {v} void
target: {@?} 0x0 (block)
argument 1: {@?} 0x0
argument 2: {@?} 0x0

Exception: decodeObjectForKey: too many nested collections
(
    0 CoreFoundation                                0x00000001ba056a5c 7519E999-1053-3367-B9D5-884
4F6D3BDC6 + 1227356
    1 libobjc.A.dylib                            0x00000001b9d7dfa4 objc_exception_throw + 56
    2 Foundation                                0x00000001ba5422d0 7B1733B1-74C9-3A33-8A58-853
B0A029826 + 2339536
    3 Foundation                                0x00000001ba54264c 7B1733B1-74C9-3A33-8A58-853
B0A029826 + 2340428
    4 Foundation                                0x00000001ba31be3c 7B1733B1-74C9-3A33-8A58-853
B0A029826 + 85564
    5 XCAutomationSupport                      0x0000000105032104 -[XCAccessibilityElement initWithCoder:] + 120
    6 Foundation                                0x00000001ba542248 7B1733B1-74C9-3A33-8A58-853
B0A029826 + 2339400
    7 Foundation                                0x00000001ba54264c 7B1733B1-74C9-3A33-8A58-853
B0A029826 + 2340428
    8 Foundation                                0x00000001ba31ba7c 7B1733B1-74C9-3A33-8A58-853
B0A029826 + 84604
    9 XCAutomationSupport                      0x000000010501bedc -[XCElementSnapshot initWithCoder:] + 120
    10 Foundation                               0x00000001ba542248 7B1733B1-74C9-3A33-8A58-853
B0A029826 + 2339400
    11 Foundation                               0x00000001ba5426a4 7B1733B1-74C9-3A33-8A58-853
B0A029826 + 2340516
    12 Foundation                               0x00000001ba559e9c 7B1733B1-74C9-3A33-8A58-853
B0A029826 + 2436764
    13 Foundation                               0x00000001ba328024 7B1733B1-74C9-3A33-8A58-853
B0A029826 + 135204
    14 Foundation                               0x00000001ba331470 7B1733B1-74C9-3A33-8A58-853
B0A029826 + 173168
    15 Foundation                               0x00000001ba542248 7B1733B1-74C9-3A33-8A58-853
B0A029826 + 2339400
    16 Foundation                               0x00000001ba54264c 7B1733B1-74C9-3A33-8A58-853
B0A029826 + 2340428
    17 Foundation                               0x00000001ba31be3c 7B1733B1-74C9-3A33-8A58-853
B0A029826 + 85564
    18 XCAutomationSupport                      0x000000010501c738 -[XCElementSnapshot initWithCoder:] + 2260
    19 Foundation                               0x00000001ba542248 7B1733B1-74C9-3A33-8A58-853
B0A029826 + 2339400

```

```

20 Foundation 0x00000001ba5426a4 7B1733B1-74C9-3A33-8A58-853
B0A029826 + 2340516
21 Foundation 0x00000001ba559e9c 7B1733B1-74C9-3A33-8A58-853
B0A029826 + 2436764
22 Foundation 0x00000001ba328024 7B1733B1-74C9-3A33-8A58-853
B0A029826 + 135204
23 Foundation 0x00000001ba331470 7B1733B1-74C9-3A33-8A58-853
B0A029826 + 173168
24 Foundation 0x00000001ba542248 7B1733B1-74C9-3A33-8A58-853
B0A029826 + 2339400
25 Foundation 0x00000001ba54264c 7B1733B1-74C9-3A33-8A58-853
B0A029826 + 2340428
26 Foundation 0x00000001ba31be3c 7B1733B1-74C9-3A33-8A58-853
B0A029826 + 85564

```

且重启WebDriverAgent也没用。

只能去iPhone中卸载掉之前的WebDriverAgentRunner-Runner后

重新安装WebDriverAgentRunner-Runner，才能继续使用。

但是本身上述页面，获取源码就导致崩溃，则无法解决。

经调试找到根本原因是：

```
refer/WebDriverAgent/WebDriverAgentLib/Utilities/FBXCodeCompatibility.m
```

中的

```

+ (XCElementSnapshot *)fb_cachedSnapshot
{
    static dispatch_once_t onceToken;
    static BOOL isUniqueMatchingSnapshotAvailable;
    dispatch_once(&onceToken, ^{
        isUniqueMatchingSnapshotAvailable = [self respondsToSelector @selector(uniqueMatchingSnapshotWithError:)];
    });
    ...
}

```

的uniqueMatchingSnapshotWithError

-» 属于Apple自己的API的bug，无法解决

详见：

【规避解决】 XCode实时调试NSXPCCConnection的
`_XCT_fetchSnapshotForElement:attributes:parameters:reply`错误

【规避解决】 WebDriverAgent获取页面源码报错：`xpc.exceptions NSXPCCConnection com.apple.testmanagerd _XCT_fetchSnapshotForElement`

获取源码速度非常慢

之前是：

【已解决】 wda用source()获取页面源码xml速度极其慢

算是：从30秒左右，不知道做了啥，变成了，优化成了，10秒多

期间：

合并了最新的WebDriverAgent的代码：

- 【已解决】 把旧版WebDriverAgent自己优化改动合并到最新版代码中
- 【已解决】 验证最新WebDriverAgent代码功能上是否正常
- 【已解决】 XCode编译最新版WebDriverAgent

也并没有解决 获取源码速度慢的问题

后来是：

- 【已解决】 用XCode实时调试WebDriverAgent希望找到并解决获取页面源码慢的原因
- 【已解决】 尝试解决facebook-wda和WebDriverAgent的获取源码很慢的原因
- 【未解决】 WebDriverAgent和wda获取源码提速：尝试shouldLoadSnapshotWithAttributes参数
- 【未解决】 调节Appium的Capability的参数去提高facebook-wda和WebDriverAgent获取源码的速度
- 【未解决】 WebDriverAgent获取源码慢尝试调节参数：
shouldUseTestManagerForVisibilityDetection
- 【已解决】 Xcode调试WebDriverAgent研究fb_waitUntilSnapshotIsStable含义希望提高获取源码速度
- 【已解决】 WebDriverAgent报错：Internal error Error Domain com.apple.dt.xctest.automation-support.error Code 5 Error kAXErrorServerNotFound getting snapshot for element
- 【已解决】 WebDriverAgent中fb_waitUntilSnapshotIsStable的作用和含义即为何加上

最终是：

- 【已解决】 WebDriverAgent获取源码慢尝试调节参数：FB_ANIMATION_TIMEOUT

解决了：

从10秒多，优化成，1~5秒左右，对于个别页面元素多时，才需要10秒+

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2021-04-11 11:34:13

常用代码段

在折腾 `facebook-wda` 期间，把各种常用的功能封装成了函数，现整理如下供参考：

`get_cmd_lines` 执行命令返回结果

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

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

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

`multipleRetry` 多次尝试执行某个函数直到成功

```
def multipleRetry(functionInfoDict, maxRetryNum=5, sleepInterval=0.1, isShowErrWhenFail=True):
    """
    do something, retry mutiple time if fail

    Args:
        functionInfoDict (dict): function info dict contain functionCallback and [optional]
    ] functionParaDict
        maxRetryNum (int): max retry number
        sleepInterval (float): sleep time of each interval when fail
        isShowErrWhenFail (bool): show error when fail if true
    Returns:
        bool
    Raises:
    """
    doSuccess = False
```

```

functionCallback = functionInfoDict["functionCallback"]
functionParaDict = functionInfoDict.get("functionParaDict", None)

curRetryNum = maxRetryNum
while curRetryNum > 0:
    if functionParaDict:
        doSuccess = functionCallback(*functionParaDict)
    else:
        doSuccess = functionCallback()

    if doSuccess:
        break

    time.sleep(sleepInterval)
    curRetryNum -= 1

    if not doSuccess:
        if isShowErrWhenFail:
            functionName = str(functionCallback)
            # '<bound method DevicesMethods.switchToAppStoreSearchTab of <src.AppCrawler.AppCrawler object at 0x1053abe80>'
            logging.error("Still fail after %d retry for %s", maxRetryNum, functionName)
return doSuccess

```

注：最新版本代码以及具体解释，详见：

[通用逻辑 · Python常用代码段](#)

isPageHasNavBar_iOS 是否包含导航栏

```

def isPageHasNavBar_iOS(self, page):
    """Check whether current page has XCUIElementTypeNavigationBar"""
    hasNavBar = False
    naviBarName = ""
    """
    has:
        <XCUIElementTypeNavigationBar type="XCUIElementTypeNavigationBar" name="微信" enabled="true" visible="true" x="0" y="20" width="414" height="44">
            <XCUIElementTypeOther type="XCUIElementTypeOther" name="微信," label="微信," enabled="true" visible="true" x="206" y="24" width="2" height="36"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="更多" label="更多" enabled="true" visible="true" x="338" y="20" width="56" height="44"/>
        </XCUIElementTypeNavigationBar>

        <XCUIElementTypeNavigationBar type="XCUIElementTypeNavigationBar" name="通讯录" enabled="true" visible="true" x="0" y="20" width="414" height="44">
            <XCUIElementTypeOther type="XCUIElementTypeOther" name="通讯录," label="通讯录," enabled="true" visible="true" x="206" y="24" width="2" height="36"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="添加朋友" label="添加朋友" enabled="true" visible="true" x="354" y="20" width="40" height="44"/>
        </XCUIElementTypeNavigationBar>
    """

```

```

        <XCUIElementTypeNavigationBar type="XCUIElementTypeNavigationBar" name="公众号"
enabled="true" visible="true" x="0" y="20" width="414" height="44">
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="返回" label="返回"
" enabled="true" visible="true" x="20" y="20" width="30" height="44"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" name="公众号," label="公
众号," enabled="true" visible="true" x="206" y="24" width="2" height="36"/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="添加" label="添加"
" enabled="true" visible="true" x="354" y="20" width="40" height="44"/>
        </XCUIElementTypeNavigationBar>

        某个公众号: 动卡空间
        <XCUIElementTypeNavigationBar type="XCUIElementTypeNavigationBar" name="动卡空
间" enabled="true" visible="true" x="0" y="20" width="414" height="44">
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="返回" label="返回"
" enabled="true" visible="true" x="20" y="20" width="30" height="44"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" name="动卡空间," label="动
卡空间," enabled="true" visible="true" x="206" y="24" width="2" height="36"/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="聊天详情" label="聊
天详情" enabled="true" visible="true" x="354" y="20" width="40" height="44"/>
        </XCUIElementTypeNavigationBar>
    not:
        <XCUIElementTypeNavigationBar type="XCUIElementTypeNavigationBar" name="通讯录"
enabled="true" visible="true" x="0" y="20" width="414" height="44">
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="返回" label="返回"
" enabled="true" visible="true" x="20" y="20" width="30" height="44"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
false" x="206" y="24" width="2" height="36"/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="搜索" label="搜索"
" enabled="true" visible="true" x="306" y="20" width="40" height="44"/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="更多" label="更多"
" enabled="true" visible="true" x="354" y="20" width="40" height="44"/>
        </XCUIElementTypeNavigationBar>
"""

soup = CommonUtils.xmlToSoup(page)
foundNaviBar = soup.find(
    'XCUIElementTypeNavigationBar',
    attrs={"type": "XCUIElementTypeNavigationBar", "enabled": "true"},
)
if foundNaviBar:
    # maybeFakeNaviBarName = foundNaviBar.attrs["name"]
    maybeFakeNaviBarName = foundNaviBar.attrs.get("name")
    typeOtherNameP = re.compile("%s,?" % maybeFakeNaviBarName)
    foundTypeOther = foundNaviBar.find("XCUIElementTypeOther", attrs={"type": "XCUIEle
mentTypeOther", "name": typeOtherNameP})
    if foundTypeOther:
        hasNaviBar = True
        naviBarName = maybeFakeNaviBarName

return hasNaviBar, naviBarName

```

调用:

```

OfflinePageNavBarNameList = [
    "微信",
    "通讯录",
    "公众号",
]
hasNavBar, naviBarName = self.isPageHasNavBar_iOS(page)

```

获取页面源码getCurPageSource_iOS

```

def getCurPageSource_iOS(self, sourceFormat="xml"):
    """Get iOS current page source of xml/json

Args:
    sourceFormat (str): source format: xml/json
Returns:
    str
Raises:
    """
    logging.info("start get iOS page source")
    pageSource = ""
    beforeGetTime = time.time()
    if sourceFormat == "xml":
        curPageXml = self.wdaClient.source() # format XML
        pageSource = curPageXml
    elif sourceFormat == "json":
        curPageJson = self.wdaClient.source(accessible=True) # default false, format JSON
        pageSource = curPageJson
    else:
        logging.error("Unsupported source format: %s", sourceFormat)
    afterGetTime = time.time()
    getSourceTime = afterGetTime - beforeGetTime
    logging.info("Cost %.2f seconds to get iOS page source", getSourceTime)
    return pageSource

```

调用：

```

def getCurPageSource(self):
    if self.isAndroid:
        return self.getCurPageSource_Android()
    elif self.isiOS:
        return self.getCurPageSource_iOS()

```

search_iOS 点击弹出的键盘中的搜索按钮 触发搜索

```

def search_iOS(self, wait=1):
    # 触发点击搜索按钮
    foundAndClickedDoSearch = False
    # <XCUIElementTypeButton type="XCUIElementTypeButton" name="Search" label="Search" ena

```

```

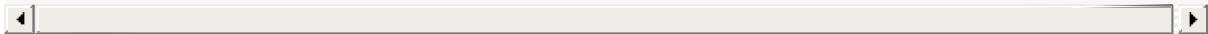
bled="true" visible="true" x="281" y="620" width="94" height="47"/>
# <XCUIElementTypeButton type="XCUIElementTypeButton" name="Search" label="搜索" enable
d="true" visible="true" x="309" y="685" width="103" height="50"/>
# searchButtonQuery = {"name": "Search"}
searchButtonQuery = {"name": "Search", "type": "XCUIElementTypeButton"}
# Note: occasionally not found Search, change to find multiple time to avoid this kind
of case
MaxRetryNumber = 5
curRetryNumber = MaxRetryNumber
while curRetryNumber > 0:
    foundAndClickedDoSearch = self.findAndClickElement(searchButtonQuery, timeout wait)

    if foundAndClickedDoSearch:
        break

    curRetryNumber -= 1

    if not foundAndClickedDoSearch:
        logging.error("Not found and/or clicked for %s", searchButtonQuery)
return foundAndClickedDoSearch

```



调用举例：

```
isSearchOk = self.search_iOS(wait 0.2)
```

wait_element_setText_iOS 给元素输入文字，设置值

```

def wait_element_setText_iOS(self, query, text, isNeedClear True):
    """iOS set element text

Args:
    query (dict): wda element query
    text (str): new text to set
    isNeedClear (bool): before set new text, is need clear current value or not
Returns:
Raises:
"""

isInputOk = False
isFound, respInfo = self.findElement(query query)
logging.debug("isFound=%s, respInfo=%s", isFound, respInfo)
if isFound:
    curElement = respInfo
    if isNeedClear:
        # before set new value, clear current value
        self.iOSClearText(curElement)
    curElement.set_text(text)
    logging.info("has input text: %s", text)
    isInputOk = True
return isInputOk

```

相关函数：

```
def iOSClearText(self, curElement):
    """iOS clear current element's text value
    Note: clear_text not working, so need use other workaround to do clear text

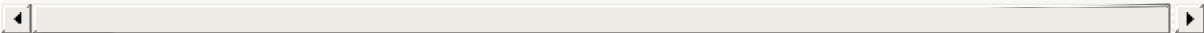
    Args:
        curElement (Element): wda element
    Returns:
    Raises:
    """
    # curElement.click()
    # curElement.clear_text()
    # curElement.tap_hold(2.0) # then try select All -> Delete
    backspaceChar = '\b'
    maxDeleteNum = 50
    curElement.set_text(maxDeleteNum * backspaceChar)
    return
```

调用举例：

(1)

对于xml

```
<XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="0" y="269" width="414" height="46">
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="269" width="414" height="1"/>
    <XCUIElementTypeTextField type="XCUIElementTypeTextField" value="your_auto_proxy_url" name="URL" label="URL" enabled="true" visible="true" x="72" y="281" width="314" height="21" />
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="313" width="414" height="2"/>
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="15" y="313" width="305" height="1"/>
</XCUIElementTypeCell>
```



调用：

```
newUrlValue = newAutoProxyValue["url"]
parentCellClassChain = "/XCUIElementTypeCell[`rect.x = 0 AND rect.width = %d`]" % self.x
urlFieldQuery = {"type": "XCUIElementTypeTextField", "name": "URL", "enabled": "true"}
urlFieldQuery["parent_class_chains"] = [parentCellClassChain]
# foundUrl, respInfo = self.findElement(urlFieldQuery, timeout=0.1)
# if not foundUrl:
#     return False
isInputUrlOk = self.wait_element_setText_iOS(urlFieldQuery, newUrlValue)
```

(2)

```
newAuthUserValue = newManualProxyValue["authUser"]
userFieldQuery = {"type": "XCUIElementTypeTextField", "name": "用户名", "enabled": "true"}
userFieldQuery["parent_class_chains"] = [ parentCellClassChain ]
isInputUserOk = self.wait_element_setText_iOS(userFieldQuery, newAuthUserValue)
```

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09 10:13:38

wda

此处整理和 facebook-wda 本身相关的，常用的代码。

init_device_driver_iOS wda基本的初始化

```

def init_device_driver_iOS(self):
    # init facebook-wda
    # wda.HTTP_TIMEOUT = self.config["WdaHttpTimeout"]
    wda.HTTP_TIMEOUT = self.config["wda"]["http"]["timeout"]
    # wda.DEBUG = self.config["WdaDebug"]
    wda.DEBUG = self.config["wda"]["debug"]

    # self.wdaClient = wda.Client('http://localhost:8100')
    # wdaServerUrl = 'http://%s:%s' % (self.config["WdaServerHost"], self.config["WdaServerPort"])
    wdaServerUrl = 'http://%s:%s' % (self.config["wda"]["server"]["host"], self.config["wda"]["server"]["port"])
    logging.info("wdaServerUrl=%s", wdaServerUrl) # 'http://localhost:8100'
    self.wdaClient = wda.Client(wdaServerUrl)
    logging.info("self.wdaClient=%s", self.wdaClient)

    self.curSession = self.wdaClient

    # init settings
    newSettings = {
        # "snapshotTimeout": self.config["WdaSnapshotTimeout"],
        # "screenshotQuality": self.config["WdaScreenQuality"],
        # "mjpecfgScalingFactor": self.config["WdaScalingFactor"],
        # "includeNonModalElements": self.config["WdaIncludeNonModalElements"],
        # "shouldUseCompactResponses": self.config["WdaShouldUseCompactResponses"],
        # "elementResponseAttributes": self.config["WdaElementResponseAttributes"],
        # }
    newSettings = [
        "snapshotTimeout": self.config["wda"]["settings"]["SnapshotTimeout"],
        "screenshotQuality": self.config["wda"]["settings"]["ScreenQuality"],
        "mjpecfgScalingFactor": self.config["wda"]["settings"]["ScalingFactor"],
        "includeNonModalElements": self.config["wda"]["settings"]["IncludeNonModalElements"]
    ],
    "shouldUseCompactResponses": self.config["wda"]["settings"]["ShouldUseCompactResponses"],
    "elementResponseAttributes": self.config["wda"]["settings"]["ElementResponseAttributes"],
    }
    respNewSettings = self.curSession.set_settings(newSettings)
    logging.debug("respNewSettings=%s", respNewSettings)

```

processWdaResponse 处理(post等返回的response) 返回数据

```

def processWdaResponse(self, wdaResponse):
    """Process common wda (http post) response

    Args:
        bool, ?/dict:
            true, response value
            false, error info dict
    Returns:
        Raises:
    """
    isRespOk = False
    respInfo = None
    logging.debug("wdaResponse=%s", wdaResponse)
    respStatus = wdaResponse.status
    respValue = wdaResponse.value
    respSessionId = wdaResponse.sessionId
    logging.debug("respStatus=%s, respValue=%s, respSessionId", respStatus, respValue, respSessionId)
    if respStatus == 0:
        isRespOk = True
        respInfo = respValue
    else:
        isRespOk = False
        errInfo = {
            "status": respStatus,
            "value": respValue,
            "sessionId": respSessionId,
        }
        respInfo = errInfo
    return isRespOk, respInfo

```

调用举例：

(1)

```

curAppState = self.wdaClient.app_state(appBundleId)
isGetOk, respInfo = self.processWdaResponse(curAppState)

```

(2)

```

terminateResp = self.wdaClient.app_terminate(appBundleId)
logging.debug("terminateResp=%s", terminateResp)
# respStatus = resp.status
# respValue = resp.value
# respSessionId = resp.sessionId
# logging.info("respStatus=%s, respValue=%s, respSessionId", respStatus, respValue, respSessionId)
# if respStatus == 0:
#     isTerminalOk = True

```

```
#     respInfo = None
# else:
#     errInfo = {
#         "status": respStatus,
#         "value": respValue,
#     }
#     respInfo = errInfo
isTerminalOk, respInfo = self.processWdaResponse(terminateResp)
```

(3)

```
launchResp = self.wdaClient.app_launch(appBundleId)
isLaunchOk, respInfo = self.processWdaResponse(launchResp)
```

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09 10:13:38

crifan版wda

此处调试期间，已经基于 v0.7.2 的原版openatx的facebook-wda做了不少改动，以便于支持很多细节功能，和修复了一些bug。

有需要的，可以下载下面源码，然后替换已安装好的 wda

注：已安装好的wda的位置，此处

是 /Users/limao/.pyenv/versions/3.8.0/Python.framework/Versions/3.8/lib/python3.8/site-packages/wda/__init__.py ，供参考。

crifan版wda源码

- 在线下载：
 - [crifan版wda __init__.py](#)
- 直接贴出源码：

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# Updated: 20200609 17:01
# Editor: Crifan Li

from __future__ import print_function, unicode_literals

import base64
import copy
import functools
import io
import json
import os
import re
import threading
import time
from collections import defaultdict, namedtuple

import requests
import retry
import six
from six.moves import urllib
try:
    from functools import cached_property # Python3.8+
except ImportError:
    from cached_property import cached_property

from . import xcui_element_types

import logging
from enum import Enum
```

```

urlparse = urllib.parse urlparse
_urljoin = urllib.parse urljoin

if six.PY3:
    from functools import reduce

DEBUG = False
# to change to logging -> later can use colorful logging
print = logging.debug

HTTP_TIMEOUT = 60.0 # unit second

# RETRY_INTERVAL = 0.01
RETRY_INTERVAL = 0.1

LANDSCAPE = 'LANDSCAPE'
PORTRAIT = 'PORTRAIT'
LANDSCAPE_RIGHT = 'UIA_DEVICE_ORIENTATION_LANDSCAPERIGHT'
PORTRAIT_UPSIDEDOWN = 'UIA_DEVICE_ORIENTATION_PORTRAIT_UPSIDEDOWN'

alert_callback = None

JSONDecodeError = json.decoder.JSONDecodeError if hasattr(
    json.decoder, "JSONDecodeError") else ValueError

#####
# Definition
#####

# https://developer.apple.com/documentation/uikit/uidevice/1620042-batterylevel?language=objc
class BatteryState(Enum):
    Unknown = 0
    Unplugged = 1
    Charging = 2
    Full = 3

# https://developer.apple.com/documentation/xctest/xctimagequality?language=objc
class ScreenshotQuality(Enum):
    Original = 0 # lossless PNG image
    Medium = 1 # high quality lossy JPEG image
    Low = 2 # highly compressed lossy JPEG image

# https://developer.apple.com/documentation/xctest/xcuiapplicationstate?language=objc
class ApplicationState(Enum):
    Unknown = 0
    NotRunning = 1
    RunningBackgroundSuspended = 2
    RunningBackground = 3
    RunningForeground = 4

class WDAError(Exception):

```

```

""" base wda error """

class WDARequestError(WDAError):
    def __init__(self, status, value):
        self.status = status
        self.value = value

    def __str__(self):
        return 'WDARequestError(status=%d, value=%s)' % (self.status,
                                                          self.value)

class WDADeleteNotFoundError(WDAError):
    """ response body is empty """

class WDAElementNotFoundError(WDAError):
    """ element not found """

class WDAElementNotDisappearError(WDAError):
    """ element not disapper """

#####
# Utils Functions
#####

def convert(dictionary):
    """
    Convert dict to namedtuple
    """
    return namedtuple('GenericDict', list(dictionary.keys()))(**dictionary)

def urljoin(urls):
    """
    The default urlparse.urljoin behavior look strange
    Standard urlparse.urljoin('http://a.com/foo', '/bar')
    Expect: http://a.com/foo/bar
    Actually: http://a.com/bar

    This function fix that.
    """
    return reduce(_urljoin, [u.strip('/') + '/' for u in urls if u.strip('/')],
                 '').rstrip('/')

def roundint(i):
    return int(round(i, 0))

```

```

def namedlock(name):
    """
    Returns:
        threading.Lock
    """
    if not hasattr(namedlock, 'locks'):
        namedlock.locks = defaultdict(threading.Lock)
    return namedlock.locks[name]

def httpdo(url, method="GET", data=None):
    """
    thread safe http request
    """
    p = urlparse(url)
    with namedlock(p.scheme + "://" + p.netloc):
        return _unsafe_httpdo(url, method, data)

def _unsafe_httpdo(url, method='GET', data=None):
    """
    Do HTTP Request
    """
    start = time.time()
    if DEBUG:
        body = json.dumps(data) if data else ''
        print("Shell: curl -X {method} -d '{body}' '{url}'".format(
            method=method.upper(), body=body or '', url=url))

    try:
        response = requests.request(method,
                                     url,
                                     json=data,
                                     timeout=HTTP_TIMEOUT)
    except (requests.exceptions.ConnectionError,
            requests.exceptions.ReadTimeout) as e:
        raise

    if DEBUG:
        ms = (time.time() - start) * 1000
        print('Return {:.0f}ms: {}'.format(ms, response.text))

    try:
        retjson = response.json()
        retjson['status'] = retjson.get('status', 0)
        r = convert(retjson)
        if r.status != 0:
            raise WDAResponseError(r.status, r.value)
        if isinstance(r.value, dict) and r.value.get("error"):
            raise WDAResponseError(100, r.value["error"])
    except JSONDecodeError:
        raise WDAResponseError(100, "JSON Decode Error")
    return r

```

```

    if response.text == "":
        raise WDAEmptyResponseError(method, url, data)
    raise WDAError(method, url, response.text)

#####
# Main
#####

class HTTPClient(object):
    def __init__(self, address, alert_callback=None):
        """
        Args:
            address (string): url address eg: http://localhost:8100
            alert_callback (func): function to call when alert popup
        """
        self.address = address
        self.alert_callback = alert_callback

    def new_client(self, path):
        return HTTPClient(
            self.address.rstrip('/') + '/' + path.lstrip('/'),
            self.alert_callback)

    def fetch(self, method, url, data=None):
        return self._fetch_no_alert(method, url, data)
        # return httpdo(urljoin(self.address, url), method, data)

    def _fetch_no_alert(self, method, url, data=None, depth=0):
        target_url = urljoin(self.address, url)
        try:
            return httpdo(target_url, method, data)
        except WDAResponseError as err:
            if depth >= 10:
                raise
            if err.status != 26: # alert status code
                raise
            if not callable(self.alert_callback):
                raise
            self.alert_callback()
            return self._fetch_no_alert(method, url, data, depth+1)

    def __getattr__(self, key):
        """ Handle GET,POST,DELETE, etc ... """
        return functools.partial(self.fetch, key)

class Rect(object):
    def __init__(self, x, y, width, height):
        self.x = x
        self.y = y
        self.width = width

```

```

        self.height = height

    def __str__(self):
        return 'Rect(x={x}, y={y}, width={w}, height={h})'.format(
            x self.x, y self.y, w self.width, h self.height)

    def __repr__(self):
        return str(self)

    # @property
    @cached_property
    def center(self):
        if DEBUG:
            print("calc center position")
        return namedtuple('Point', ['x', 'y'])(self.x + self.width / 2,
                                                self.y + self.height / 2)

    @property
    def origin(self):
        return namedtuple('Point', ['x', 'y'])(self.x, self.y)

    @property
    def left(self):
        return self.x

    @property
    def top(self):
        return self.y

    @property
    def right(self):
        return self.x + self.width

    @property
    def bottom(self):
        return self.y + self.height

    @property
    def x0(self):
        return self.x

    @property
    def y0(self):
        return self.y

    @property
    def x1(self):
        return self.x + self.width

    @property
    def y1(self):
        return self.y + self.height

```

```

@property
def centerX(self):
    return self.center[0]
# return self.center["x"]

@property
def centerY(self):
    return self.center[1]
# return self.center["y"]

@property
def isZero(self):
    isXZero = self.x == 0
    isYZero = self.y == 0
    isWidthZero = self.width == 0
    isHeightZero = self.height == 0
    isAllZero = isXZero and isYZero and isWidthZero and isHeightZero
    return isAllZero


class Client(object):
    def __init__(self, url=None, _session_id=None):
        """
        Args:
            target (string): the device url

            If target is empty, device url will set to env-var "DEVICE_URL" if defined else set to "http://localhost:8100"
        """
        if not url:
            url = os.environ.get('DEVICE_URL', 'http://localhost:8100')
        assert re.match(r'^https?://', url), "Invalid URL: %r" % url

        self.http = HTTPClient(url)

        # Session variable
        self._session_id = _session_id
        self._timeout = 30.0
        self._target = None

    def wait_ready(self, timeout=120):
        """
        wait until WDA back to normal

        Returns:
            bool (if wda works)
        """
        deadline = time.time() + timeout
        while time.time() < deadline:
            try:
                self.status()
                return True
            except:

```

```

        time.sleep(2)
    return False

@retry.retry(exceptions.WDAEmptyResponseError, tries=3, delay=2)
def status(self):
    res = self.http.get('status')
    sid = res.sessionId
    res.value['sessionId'] = sid
    return res.value

def home(self):
    """Press home button"""
    return self.http.post('/wda/homescreen')

def healthcheck(self):
    """Hit healthcheck"""
    return self.http.get('/wda/healthcheck')

def locked(self):
    """ returns locked status, true or false """
    return self.http.get("/wda/locked").value

def lock(self):
    return self.http.post('/wda/lock')

def unlock(self):
    """ unlock screen, double press home """
    return self.http.post('/wda/unlock')

def app_current(self):
    """
    Returns:
        dict, eg:
    {
        "running": true,
        "state": 4,
        "generation": 0,
        "processArguments": {
            "env": {},
            "args": []
        },
        "title": "",
        "bundleId": "com.tencent.xin",
        "label": "微信",
        "path": "",
        "name": "",
        "pid": 1357
    }
    """
    return self.http.get("/wda/activeAppInfo").value

def source(self, format='xml', accessible=False):
    """
    """

```

```

Args:
    format (str): only 'xml' and 'json' source types are supported
    accessible (bool): when set to true, format is always 'json'
    """
    if accessible:
        return self.http.get('/wda/accessibleSource').value
    return self.http.get('source?format=' + format).value

def screenshot(self, png_filename=None, format='pillow'):
    """
    Screenshot with PNG format

    Args:
        png_filename(string): optional, save file name
        format(string): return format, "raw" or "pillow" (default)

    Returns:
        PIL.Image or raw png data

    Raises:
        WDAResponseError
    """
    value = self.http.get('screenshot').value
    raw_value = base64.b64decode(value)
    png_header = b"\x89PNG\r\n\x1a\n"
    if not raw_value.startswith(png_header) and png_filename:
        raise WDAResponseError(1, "screenshot png format error")

    if png_filename:
        with open(png_filename, 'wb') as f:
            f.write(raw_value)

    if format == 'raw':
        return raw_value
    elif format == 'pillow':
        from PIL import Image
        buff = io.BytesIO(raw_value)
        return Image.open(buff)
    else:
        raise ValueError("unknown format")

def session(self,
            bundle_id=None,
            arguments=None,
            environment=None,
            alert_action=None):
    """
    Args:
        - bundle_id (str): the app bundle id
        - arguments (list): ['-u', 'https://www.google.com/ncr']
        - environment (dict): {"KEY": "VAL"}
        - alert_action (str): "accept" or "dismiss"

```

```

WDA Return json like

{
    "value": {
        "sessionId": "69E6FDBA-8D59-4349-B7DE-A9CA41A97814",
        "capabilities": {
            "device": "iphone",
            "browserName": "部落冲突",
            "sdkVersion": "9.3.2",
            "CFBundleIdentifier": "com.supercell.magic"
        }
    },
    "sessionId": "69E6FDBA-8D59-4349-B7DE-A9CA41A97814",
    "status": 0
}

To create a new session, send json data like

{
    "desiredCapabilities": {
        "bundleId": "your-bundle-id",
        "app": "your-app-path"
        "shouldUseCompactResponses": (bool),
        "shouldUseTestManagerForVisibilityDetection": (bool),
        "maxTypingFrequency": (integer),
        "arguments": (list(str)),
        "environment": (dict: str->str)
    },
}
"""

if bundle_id is None:
    return self

if arguments and type(arguments) is not list:
    raise TypeError('arguments must be a list')

if environment and type(environment) is not dict:
    raise TypeError('environment must be a dict')

capabilities = {
    'bundleId': bundle_id,
    'arguments': arguments,
    'environment': environment,
    'shouldWaitForQuiescence': False,
    # In the latest appium/WebDriverAgent, set shouldWaitForQuiescence to True will
    # stuck
    # 'useJSONSource': True,
    # 'simpleIsVisibleCheck': True,
    # 'shouldUseTestManagerForVisibilityDetection': True,
}
# Remove empty value to prevent WDAResponseError
for k in list(capabilities.keys()):
    if capabilities[k] is None:

```

```

        capabilities.pop(k)

    if alert_action:
        assert alert_action in ["accept", "dismiss"]
        capabilities["defaultAlertAction"] = alert_action

    data = {
        'desiredCapabilities': capabilities, # For old WDA
        "capabilities": {
            "alwaysMatch": capabilities, # For recent WDA 2019/08/28
        }
    }
    if DEBUG:
        print("data=%s" % data)
    try:
        res = self.http.post('session', data)
        # if DEBUG:
        #     # print("res=%s" % res)
        #     # respJson = res.json()
        #     # print("respJson=%s" % respJson)
        #     respDict = dict(res)
        #     print("respDict=%s" % respDict)
    except WDADebugResponseError:
        """ when there is alert, might be got empty response
        use /wda/apps/state may still get sessionId
        """
        res = self.session().app_state(bundle_id)
        if res.value != 4:
            raise
    return Client(self.http.address, _session_id=res.sessionId)

#####
##### Session methods and properties #####
#####

def __str__(self):
    # return 'wda.Client (sessionId=%s)' % self._sid
    return 'wda.Client (sessionId=%s)' % self.__session_id

def __enter__(self):
    """
    Usage example:
    with c.session("com.example.app") as app:
        # do something
    """
    return self

def __exit__(self, exc_type, exc_value, traceback):
    self.close()

@property
def id(self):
    return self._session_id

```

```

@property
def _session_id(self) -> str:
    if self._session_id:
        return self._session_id
    return self.status()['sessionId']

@property
def _session_http(self) -> HTTPClient:
    return self.http.new_client("session/" + self._session_id)

@cached_property
def scale(self):
    """
    UIKit scale factor

    Refs:
        https://developer.apple.com/library/archive/documentation/DeviceInformation/Reference/iOSDeviceCompatibility/Displays/Displays.html
        There is another way to get scale
            self.http.get("/wda/screen").value returns {"statusBarSize": {'width': 320, 'height': 20}, 'scale': 2}
    """
    v = max(self.screenshot().size) / max(self.window_size())
    if DEBUG:
        print("v=%s" % v)
    return round(v)

@cached_property
def bundle_id(self):
    """ the session matched bundle id """
    v = self._session_http.get("/").value
    return v['capabilities'].get('CFBundleIdentifier')

def implicitly_wait(self, seconds):
    """
    set default element search timeout
    """
    assert isinstance(seconds, (int, float))
    self.__timeout = seconds

def battery_info(self):
    """
    Returns dict:
        eg: {"level": 1, "state": 2}
        level: 0 ~ 1.0, battery electricity percent
        state: unknown=0, unplugged=1, charging=2, full=3
        https://developer.apple.com/documentation/uikit/uidevice/batterystate
    """
    return self._session_http.get("/wda/batteryInfo").value

def matchTouchID(self):
    postResp = self._session_http.post("/wda/touch_id", {
        "match": 1,

```

```

        })
    respValue = postResp.value
    if DEBUG:
        print("/wda/touch_id: respValue=%s" % respValue)
    return respValue

    def device_info(self):
        """
        Returns dict:
        eg: {'currentLocale': 'zh_CN', 'timeZone': 'Asia/Shanghai'}
        """
        return self._session_http.get("/wda/device/info").value

    def screen_info(self):
        """get screen info
           https://developer.apple.com/library/archive/documentation/DeviceInformation/Reference/iOSDeviceCompatibility/Displays/Displays.html

        Returns dict:
        eg:
            {'statusBarSize': {'width': 375, 'height': 20}, 'scale': 2}
            -> updated wda code, can return more info:
            {'statusBarSize': {'width': 375, 'height': 20}, 'nativeScale': 2.88, 'nativeBounds': {'y': 0, 'x': 0, 'width': 921, 'height': 1382}, 'bounds': {'y': 0, 'x': 0, 'width': 320, 'height': 480}, 'scale': 2}
        """
        if DEBUG:
            print("self._session_http=%s" % self._session_http)

        return self._session_http.get("/wda/screen").value
        # return self.http.get("/wda/screen").value

    def window_info(self):
        """get window size info

        Returns dict:
        eg: {'width': 375, 'height': 667}
        """
        return self._session_http.get("/window/size").value
        # return self.http.get("/window/size").value

    def get_settings(self):
        resp = self._session_http.get("/appium/settings")
        respSettings = resp.value
        if DEBUG:
            # print("resp=%s" % resp) # TypeError not all arguments converted during string formatting
            print("respSettings=%s" % respSettings)
        return respSettings

    def set_settings(self, newSettings):
        postResp = self._session_http.post("/appium/settings", {
            "settings": newSettings,

```

```

        })
    respNewSettings = postResp.value
    if DEBUG:
        print("respNewSettings=%s" % respNewSettings)
    return respNewSettings

def setSnapshotTimeout(self, newSnapshotTimeout):
    """set new snapshotTimeout value for current session /appium/settings"""
    newSettings = {
        "snapshotTimeout": newSnapshotTimeout
    }
    return self.set_settings(newSettings)

def set_clipboard(self, content, content_type="plaintext"):
    """ set clipboard """
    self._session_http.post(
        "/wda/setPasteboard", {
            "content": base64.b64encode(content.encode()).decode(),
            "contentType": content_type
        })

def set_alert_callback(self, callback):
    """
    Args:
        callback (func): called when alert popup
    """

    Example of callback:

    def callback(session):
        session.alert.accept()
    """

    if callable(callback):
        self.http.alert_callback = functools.partial(callback, self)
    else:
        self.http.alert_callback = None

#Not working
#def get_clipboard(self):
#    self.http.post("/wda/getPasteboard").value

# Not working
#def siri_activate(self, text):
#    self.http.post("/wda/siri/activate", {"text": text})

def app_launch(self,
              bundle_id,
              arguments=[],
              environment={},
              wait_for_quiescence=False):
    """

    Args:
        - bundle_id (str): the app bundle id
        - arguments (list): ['-u', 'https://www.google.com/ncr']
        - enviroment (dict): {"KEY": "VAL"}
    """

```

```

        - wait_for_quiescence (bool): default False
"""

assert isinstance(arguments, (tuple, list))
assert isinstance(environment, dict)

return self._session_http.post(
    "/wda/apps/launch", {
        "bundleId": bundle_id,
        "arguments": arguments,
        "environment": environment,
        "shouldWaitForQuiescence": wait_for_quiescence,
    })
}

def app_activate(self, bundle_id):
    return self._session_http.post("/wda/apps/launch", {
        "bundleId": bundle_id,
    })

def app_terminate(self, bundle_id):
    return self._session_http.post("/wda/apps/terminate", {
        "bundleId": bundle_id,
    })

def app_state(self, bundle_id):
    """
    Returns example:
    {
        "value": 4,
        "sessionId": "0363BDC5-4335-47ED-A54E-F7CCB65C6A65"
    }

    value: enum ApplicationState
"""
    return self._session_http.post("/wda/apps/state", {
        "bundleId": bundle_id,
    })

def app_list(self):
    """
    Not working very well, only show springboard

    Returns:
        list of app
    """

    Return example:
        [{'pid': 52, 'bundleId': 'com.apple.springboard'}]
"""
    return self._session_http.get("/wda/apps/list").value

def open_url(self, url):
    """
    TODO: Never successed using before. Looks like use Siri to search.
    https://github.com/facebook/WebDriverAgent/blob/master/WebDriverAgentLib/Commands/

```

```

FBSessionCommands.m#L43
    Args:
        url (str): url

    Raises:
        WDAResponseError
    """
    return self._session_http.post('url', {'url': url})

def deactivate(self, duration):
    """Put app into background and than put it back
    Args:
        - duration (float): deactivate time, seconds
    """
    return self._session_http.post('/wda/deactivateApp', dict(duration=duration))

def tap(self, x, y):
    return self._session_http.post('/wda/tap/0', dict(x=x, y=y))

def _percent2pos(self, x, y, window_size=None):
    if DEBUG:
        print("x=%s, y=%s, window_size=%s" % (x, y, window_size))

    if any(isinstance(v, float) for v in [x, y]):
        w, h = window_size or self.window_size()
        if DEBUG:
            print("type(w)=%s" % type(w))
            print("type(h)=%s" % type(h))
            print("w=%s, h=%s" % (w, h))

        # x = int(x * w) if isinstance(x, float) else x
        # y = int(y * h) if isinstance(y, float) else y
        if isinstance(x, float):
            if (x > 0.0) and (x <= 1.0):
                convertedX = x * w
                xInt = int(convertedX)
            else:
                # consider as real float position value
                xInt = int(x)
            x = xInt

        if isinstance(y, float):
            if (y > 0.0) and (y <= 1.0):
                convertedY = y * h
                yInt = int(convertedY)
            else:
                # consider as real float position value
                yInt = int(y)
            y = yInt

    if DEBUG:
        print("type(x)=%s" % type(x))
        print("type(y)=%s" % type(y))

```

```

        print("x=%s, y=%s" % (x, y))

    assert w >= x >= 0
    assert h >= y >= 0
    return (x, y)

def click(self, x, y):
    """
    x, y can be float(percent) or int
    """
    x, y = self._percent2pos(x, y)
    return self.tap(x, y)

def double_tap(self, x, y):
    x, y = self._percent2pos(x, y)
    return self._session_http.post('/wda/doubleTap', dict(x=x, y=y))

def tap_hold(self, x, y, duration 1.0):
    """
    Tap and hold for a moment

    Args:
        - x, y(int, float): float(percent) or int(absolute coordinate)
        - duration(float): seconds of hold time

    [[FBRoute POST:@"/wda/touchAndHold"] respondWithTarget:self action:@selector(handleTouchAndHoldCoordinate:)],
    """
    x, y = self._percent2pos(x, y)
    data = {'x': x, 'y': y, 'duration': duration}
    return self._session_http.post('/wda/touchAndHold', data=data)

def swipe(self, x1, y1, x2, y2, duration 0):
    """
    Args:
        x1, y1, x2, y2 (int, float): float(percent), int(coordinate)
        duration (float): start coordinate press duration (seconds)

    [[FBRoute POST:@"/wda/dragfromtoforduration"] respondWithTarget:self action:@selector(handleDragCoordinate:)],
    """
    if any(isinstance(v, float) for v in [x1, y1, x2, y2]):
        size = self.window_size()
        x1, y1 = self._percent2pos(x1, y1, size)
        x2, y2 = self._percent2pos(x2, y2, size)

    data = dict(fromX=x1, fromY=y1, toX=x2, toY=y2, duration=duration)
    return self._session_http.post('/wda/dragfromtoforduration', data=data)

def swipe_left(self):
    w, h = self.window_size()
    return self.swipe(w, h // 2, 0, h // 2)

```

```

def swipe_right(self):
    w, h = self.window_size()
    return self.swipe(0, h // 2, w, h // 2)

def swipe_up(self):
    w, h = self.window_size()
    return self.swipe(w // 2, h, w // 2, 0)

def swipe_down(self):
    w, h = self.window_size()
    return self.swipe(w // 2, 0, w // 2, h)

@property
def orientation(self):
    """
    Return string
    One of <PORTRAIT | LANDSCAPE>
    """
    return self._session_http.get('orientation').value

@orientation.setter
def orientation(self, value):
    """
    Args:
        - orientation(string): LANDSCAPE | PORTRAIT | UIA_DEVICE_ORIENTATION_LANDSCAPE
        RIGHT | UIA_DEVICE_ORIENTATION_PORTRAIT_UPSIDEDOWN
    """
    return self._session_http.post('orientation', data={'orientation': value})

def window_size(self):
    """
    Returns:
        namedtuple: eg
        Size(width=320, height=568)
    """
    value = self._session_http.get('/window/size').value
    w = roundint(value['width'])
    h = roundint(value['height'])
    return namedtuple('Size', ['width', 'height'])(w, h)

def send_keys(self, value):
    """
    send keys, yet I know not, todo function
    """
    if isinstance(value, six.string_types):
        value = list(value)
    return self._session_http.post('/wda/keys', data={'value': value})

def keyboard_dismiss(self):
    """
    Not working for now
    """

```

```

        raise RuntimeError("not pass tests, this method is not allowed to use")
        self._session_http.post('/wda/keyboard/dismiss')

    def xpath(self, value):
        """
        For weditor, d.xpath(...)
        """
        httpclient = self._session_http.new_client('')
        return Selector(httpclient, self, xpath value)

    @property
    def alert(self):
        return Alert(self)

    def close(self): # close session
        return self._session_http.delete('/')

    def __call__(self, *args, **kwargs):
        if 'timeout' not in kwargs:
            kwargs['timeout'] = self._timeout
        return Selector(self._session_http, self, *args, **kwargs)

    @property
    def alibaba(self):
        """ Only used in alibaba company """
        try:
            import wda_taobao
            return wda_taobao.Alibaba(self)
        except ImportError:
            raise RuntimeError(
                "@alibaba property requires wda_taobao library installed")

    @property
    def taobao(self):
        try:
            import wda_taobao
            return wda_taobao.Taobao(self)
        except ImportError:
            raise RuntimeError(
                "@taobao property requires wda_taobao library installed")

Session = Client # for compatibility

class Alert(object):
    def __init__(self, client):
        self._c = client
        self.http = client.http

    @property
    def exists(self):
        try:
            self.text
        
```

```

        except WDAResponseError as e:
            if e.status != 27:
                raise
            return False
        return True

@property
def text(self):
    return self.http.get('/alert/text').value
# return self._c._session_http.get('/alert/text').value

def wait(self, timeout=20.0):
    start_time = time.time()
    while time.time() - start_time < timeout:
        if self.exists:
            return True
        time.sleep(0.2)
    return False

def accept(self):
    return self.http.post('/alert/accept')

def dismiss(self):
    return self.http.post('/alert/dismiss')

def buttons(self):
    return self._c._session_http.get('/wda/alert/buttons').value
# return self.http.get('/wda/alert/buttons').value

def click(self, button_name):
    """
    Args:
        - button_name: the name of the button
    """
    # Actually, It has no difference POST to accept or dismiss
    return self.http.post('/alert/accept', data={"name": button_name})

class Selector(object):
    def __init__(self,
                 httpclient,
                 session,
                 predicate=None,
                 id=None,
                 className=None,
                 type=None,
                 name=None,
                 nameContains=None,
                 nameMatches=None,
                 text=None,
                 textContains=None,
                 textMatches=None,
                 value=None,
                 valueContains=None,

```

```

    valueMatches=None,
    label=None,
    labelContains=None,
    visible=None,
    enabled=None,

    # https://github.com/facebookarchive/WebDriverAgent/wiki/Predicate-Query
    s-Construction-Rules
        # rect - Element's rectangle as a dictionary with the following keys: x,
        y, width, height
        rect=None,
        x=None,
        y=None,
        width=None,
        height=None,

        classChain=None,
        xpath=None,
        parent_class_chains=[],
        timeout=10.0,
        index=0):
    ...

Args:
    predicate(str): predicate string
    id(str): raw identifier
    className(str): attr of className
    type(str): alias of className
    name(str): attr for name
    nameContains(str): attr of name contains
    nameMatches(str): regex string
    text(str): alias of name
    textContains(str): alias of nameContains
    textMatches(str): alias of nameMatches
    value(str): attr of value, not used in most times
    valueContains(str): attr of value contains
    valueMatches(str): attr of value regex string
    label(str): attr for label
    labelContains(str): attr for label contains
    visible(bool): is visible
    enabled(bool): is enabled
    rect(dict): Element's rectangle as a dictionary with the following keys: x,
    y, width, height
        x(int/str): x of rect
        y(int/str): y of rect
        width(int/str): width of rect
        height(int/str): height of rect
    classChain(str): string of ios chain query, eg: **/XCUIElementTypeOther[`valu
    e BEGINSWITH 'blabla'`]
    xpath(str): xpath string, a little slow, but works fine
    timeout(float): maximum wait element time, default 10.0s
    index(int): index of founded elements

WDA use two key to find elements "using", "value"

```

```

Examples:
"using" can be one of
    "partial link text", "link text"
    "name", "id", "accessibility id"
    "class name", "class chain", "xpath", "predicate string"

predicate string support many keys
    UID,
    accessibilityContainer,
    accessible,
    enabled,
    frame,
    label,
    name,
    rect,
    type,
    value,
    visible,
    wdAccessibilityContainer,
    wdAccessible,
    wdEnabled,
    wdFrame,
    wdLabel,
    wdName,
    wdRect,
    wdType,
    wdUID,
    wdValue,
    wdVisible
    ...
if DEBUG:
    print("Selector __init__: httpclient=%s, session=%s, predicate=%s, id=%s" % (h
    ttpclient, session, predicate, id))
    print("Selector __init__: className=%s, type=%s, name=%s, nameContains=%s, nam
    eMatches=%s" % (className, type, name, nameContains, nameMatches))
    print("Selector __init__: text=%s, textContains=%s, textMatches=%s, value=%s,
    valueContains=%s, valueMatches=%s" % (text, textContains, textMatches, value, valueContains
    , valueMatches))
    print("Selector __init__: label=%s, labelContains=%s, visible=%s, value=%s, en
    abled=%s" % (label, labelContains, visible, value, enabled))
    print("Selector __init__: rect=%s, x=%s, y=%s, width=%s, height=%s" % (rect, x,
    y, width, height))
    print("Selector __init__: classChain=%s, xpath=%s, parent_class_chains=%s, tim
    eout=%s, index=%s" % (classChain, xpath, parent_class_chains, timeout, index))

    self.http = httpclient
    self.session = session

    self.predicate = predicate
    self.id = id
    self.class_name = className or type
    self.name = self._add_escape_character_for_quote_prime_character(
        name or text)

```

```

        self.name_part = nameContains or textContains
        self.name_regex = nameMatches or textMatches
        self.value = value
        self.value_part = valueContains
        self.value_regex = valueMatches
        self.label = label
        self.label_part = labelContains

        # self.enabled = enabled
        # self.visible = visible
        # self.enabled = self._toBool(enabled)
        # self.visible = self._toBool(visible)
        if enabled is not None:
            self.enabled = self._toBool(enabled)
        else:
            self.enabled = None

        if visible is not None:
            self.visible = self._toBool(visible)
        else:
            self.visible = None

        if rect:
            self.rect = rect
        else:
            self.rect = None

        rectDict = {}

        if x:
            self.x = int(x)
            rectDict["x"] = self.x
        if y:
            self.y = int(y)
            rectDict["y"] = self.y
        if width:
            self.width = int(width)
            rectDict["width"] = self.width
        if height:
            self.height = int(height)
            rectDict["height"] = self.height

        if rectDict:
            self.rect = rectDict

        self.index = index

        self.xpath = self._fix_xcui_type(xpath)
        self.class_chain = self._fix_xcui_type(classChain)
        self.timeout = timeout
        # some fixtures
        if self.class_name and not self.class_name.startswith(
            'XCUIElementType'):
    
```

```

        self.class_name = 'XCUIElementType' + self.class_name
    if self.name_regex:
        if not self.name_regex.startswith(
            '^') and not self.name_regex.startswith('.*'):
            self.name_regex = '.*' + self.name_regex
        if not self.name_regex.endswith(
            '$') and not self.name_regex.endswith('.*'):
            self.name_regex = self.name_regex + '.*'
    if DEBUG:
        print("after update: self.name_regex=%s" % self.name_regex)
    self.parent_class_chains = parent_class_chains

def _toBool(self, inputValue):
    respBool = False
    if isinstance(inputValue, bool):
        respBool = inputValue
    elif isinstance(inputValue, str):
        lowerStr = inputValue.lower()
        TrueStrList = ["true", "yes", "1"]
        if lowerStr in TrueStrList:
            respBool = True
    elif isinstance(inputValue, int):
        if inputValue >= 1:
            respBool = True

    if DEBUG:
        print("inputValue=%s -> respBool=%s" % (inputValue, respBool))

    return respBool

def _fix_xcui_type(self, s):
    if s is None:
        return
    re_element = '|'.join(xcui_element_types.ELEMENTS)
    return re.sub(r'/(^|' + re_element + ')', '/XCUIElementType\g<1>', s)

def _add_escape_character_for_quote_prime_character(self, text):
    """
    Fix for https://github.com/openatx/facebook-wda/issues/33
    Returns:
        string with properly formated quotes, or non changed text
    """
    if text is not None:
        if '"' in text:
            return text.replace('"', '\\"')
        elif "'" in text:
            return text.replace("'", "\\'")
        else:
            return text
    else:
        return text

def _wdasearch_single(self, using, value):

```

```

"""
>Returns:
    bool, str
        True, element info:
    {
        "id": "0F000000-0000-0000-D703-000000000000",
        "name": "通讯录",
        "value": None,
        "text": "通讯录",
        "label": "通讯录",
        "rect": {
            "y": 619,
            "x": 96,
            "width": 90,
            "height": 48
        },
        "type": "XCUIElementTypeButton"
    }

    False, error info:
    {
        "error" : "no such element",
        "message" : "unable to find an element using 'class chain', value
'**\\XCUIElementTypeAny[`name == '搜索``]'",
        "traceback" : "(\n\t0 WebDriverAgentLib
000100f011ec FBNoSuchElementExceptionForRequest..... 4\n)"
    }
"""

isFound, respInfo = False, "Unknown error"

postDict = {
    'using': using,
    'value': value
}

if DEBUG:
    print("postDict=%s" % postDict)

isException = False
# resp = self.http.post('/element', postDict)
try:
    resp = self.http.post('/element', postDict)
except WDAResponseError as wdaReqErr:
    isException = True
    if DEBUG:
        print("wdaReqErr=%s" % wdaReqErr)

    errCode = wdaReqErr.status
    errMsg = wdaReqErr.value
    errorInfo = {
        "error": errCode,
        "message": errMsg
}

```

```

respInfo = errorInfo

if not isException:
    # if DEBUG:
    #     print("postDict=%s -> resp=%s" % (postDict, resp))

respValue = resp.value
# if DEBUG:
#     print("respValue=%s" % respValue)

respValueKeys = respValue.keys()
hasError = "error" in respValueKeys
hasMessage = "message" in respValueKeys
isError = hasError and hasMessage

if isError:
    """
    {
        "value" : {
            "error" : "no such element",
            "message" : "unable to find an element using 'class chain', value
'**\[XCUIElementTypeAny[`name == '搜索']]''",
            "traceback" : "(\n\t0 WebDriverAgentLib 0x0000000100f011ec FB
NoSuchElementErrorResponseForRequest..... 4\n)"
        },
        "sessionId" : "DCF1A843-9FB9-4415-9A7C-8AF4ACDC3046"
    }
    """
    isFound = False
    errorInfo = respValue
    respInfo = errorInfo
else:
    isFound = True
    elementId = respValue['ELEMENT']
    # if DEBUG:
    #     print("elementId=%s" % elementId)

    """
    {
        "value" : {
            "text" : null,
            "element-6066-11e4-a52e-4f735466cecf" : "00000000-0000-0000-0000-0
00000000000",
            "label" : null,
            "attribute\[name" : null,
            "attribute\[value" : null,
            "ELEMENT" : "00000000-0000-0000-0000-000000000000"
        },
        "sessionId" : "E649E60F-16A1-48D4-8840-67387787A0A2"
    }
    """
    EmptyId = "00000000-0000-0000-0000-000000000000"

```

```

        if elementId == EmptyId:
            isFound = False
            respInfo = {
                "error" : "empty id",
                "message" : "found element but id is empty %s from %s" % (EmptyId,
str(postDict)),
            }
        else:
            """
            {
                "value" : {
                    "element-6066-11e4-a52e-4f735466cecf" : "19010000-0000-0000-A3
03-000000000000",
                    "attribute/name" : "添加",
                    "attribute/value" : null,
                    "text" : "添加",
                    "label" : "添加",
                    "rect" : {
                        "y" : 20,
                        "x" : 317,
                        "width" : 58,
                        "height" : 44
                    },
                    "type" : "XCUIElementTypeButton",
                    "name" : "XCUIElementTypeButton",
                    "ELEMENT" : "19010000-0000-0000-A303-000000000000"
                },
                "sessionId" : "DCF1A843-9FB9-4415-9A7C-8AF4ACDC3046"
            }
            """
            elementInfo = {
                "id": elementId
            }
            # respKeyList = ["type","label","name","text","rect","attribute/name",
"attribute/value"]
            # mapKeyList =  ["type","label",      "", "text","rect",           "name",
"value"]
            respKeyMap = {
                "type": "type",
                "label": "label",
                "name": None,
                "text": "text",
                "rect": "rect",
                "attribute/name": "name",
                "attribute/value": "value",
            }

            for eachKey in respValueKeys:
                if eachKey in respKeyMap.keys():
                    eachValue = respValue[eachKey]
                    mapRealKey = respKeyMap[eachKey]
                    if mapRealKey:
                        elementInfo[mapRealKey] = eachValue

```

```

        respInfo = elementInfo

    if DEBUG:
        print("isFound=%s, respInfo=%s" % (isFound, respInfo))

    # return elementId
    return isFound, respInfo

def _wdasearch(self, using, value):
    """
    Returns:
        element_ids (list(string)): example ['id1', 'id2']

    HTTP example response:
    [
        {"ELEMENT": "E2FF5B2A-DBDF-4E67-9179-91609480D80A"},
        {"ELEMENT": "597B1A1E-70B9-4CBE-ACAD-40943B0A6034"}
    ]
    """
    if DEBUG:
        print("using=%s, value=%s" % (using, value))

    element_ids = []
    # for v in self.http.post('/elements', {
    #     'using': using,
    #     'value': value
    # }).value:
    #     element_ids.append(v['ELEMENT'])
    resp = self.http.post('/elements', {
        'using': using,
        'value': value
    })
    valueList = resp.value

    if DEBUG:
        print("valueList=%s" % valueList)

    for eachValue in valueList:
        eachElement = eachValue['ELEMENT']
        element_ids.append(eachElement)

    if DEBUG:
        print("element_ids=%s" % element_ids)

    return element_ids

def _gen_class_chain(self):
    # just return if already exists predicate
    if self.predicate:
        return '/XCUIElementTypeAny[`' + self.predicate + '`]'

    qs = []
    if self.name:

```

```

        qs.append("name == '%s'" % self.name)
    if self.name_part:
        qs.append("name CONTAINS '%s'" % self.name_part)
    if self.name_regex:
        # escapedNameRegex = self.name_regex.encode('unicode_escape')
        # if DEBUG:
        #     print("self.name_regex=%s, escapedNameRegex=%s" % (self.name_regex, escapedNameRegex))
        # qs.append("name MATCHES '%s'" % escapedNameRegex)
        qs.append("name MATCHES '%s'" % self.name_regex)
        if DEBUG:
            print("after add name_regex: qs=%s" % qs)
    if self.label:
        qs.append("label == '%s'" % self.label)
    if self.label_part:
        qs.append("label CONTAINS '%s'" % self.label_part)
    if self.value:
        qs.append("value == '%s'" % self.value)
    if self.value_part:
        # qs.append("value CONTAINS '%s'" % self.value_part)
        qs.append("value CONTAINS '%s'" % self.value_part)
    if self.value_regex:
        # escapedValueRegex = self.value_regex.encode('unicode_escape')
        # if DEBUG:
        #     print("self.value_regex=%s, escapedValueRegex=%s" % (self.value_regex, escapedValueRegex))
        # qs.append("value MATCHES '%s'" % escapedValueRegex)
        qs.append("value MATCHES '%s'" % self.value_regex)
        if DEBUG:
            print("after add value_regex: qs=%s" % qs)

    # BoolTrueValue = "true"
    # BoolFalseValue = "false"
    BoolTrueValue = "1"
    BoolFalseValue = "0"

    if self.enabled is not None:
        enabledValue = BoolTrueValue if self.enabled else BoolFalseValue
        enableKey = "enabled"
        # enableKey = "wdEnabled"
        qs.append("%s == %s" % (enableKey, enabledValue))

    # 20200427: after merge latest WebDriverAgent, continue test visible
    if DEBUG:
        print("self.visible=%s" % self.visible)
    if self.visible is not None:
        visibleKey = "visible"
        # Note: WebDriverAgent says support visible, but actually not support visible
        # so temp change to (maybe similar meaning) accessible
        # 20200402: sometime use visible or accessible can NOT found element, so not use it
        # visibleKey = "accessible"
        # 20200408: debug displayed

```

```

        # visibleKey = "displayed"
        visibleValue = BoolTrueValue if self.visible else BoolFalseValue
        visibleConditionStr = "%s == %s" % (visibleKey, visibleValue)
        qs.append(visibleConditionStr)
        if DEBUG:
            print("visibleConditionStr=%s" % visibleConditionStr)
            print("qs=%s" % qs)

    if self.rect:
        rectKeyList = ["x", "y", "width", "height"]
        for eachRectKey in rectKeyList:
            if eachRectKey in self.rect.keys():
                eachRectValue = self.rect[eachRectKey]
                curMatchRule = "rect.%s == %s" % (eachRectKey, eachRectValue)
                qs.append(curMatchRule)

    predicate = ' AND '.join(qs)
    chain = '/' + (self.class_name or 'XCUIElementTypeAny')

    if predicate:
        chain = chain + '[' + predicate + ']'
    if self.index:
        chain = chain + '[%d]' % self.index

    if DEBUG:
        print("generated class chain=%s" % chain)

    return chain

def find_element_ids(self):
    elems = []
    if self.id:
        return self._wdasearch('id', self.id)
    if self.predicate:
        return self._wdasearch('predicate string', self.predicate)
    if self.xpath:
        return self._wdasearch('xpath', self.xpath)
    if self.class_chain:
        return self._wdasearch('class chain', self.class_chain)

    chain = '***' + ''.join(
        self.parent_class_chains) + self._gen_class_chain()
    if DEBUG:
        print('CHAIN: %s', chain)
    return self._wdasearch('class chain', chain)

def find_element_id(self):
    if self.id:
        return self._wdasearch_single('id', self.id)

    if self.predicate:
        if DEBUG:
            print('PREDICATE: %s', self.predicate)

```

```

        return self._wdasearch_single('predicate string', self.predicate)

    if self.xpath:
        if DEBUG:
            print('XPATH: %s', self.xpath)
        return self._wdasearch_single('xpath', self.xpath)

    if self.class_chain:
        if DEBUG:
            print('CLASS_CHAIN: %s', self.class_chain)
        return self._wdasearch_single('class chain', self.class_chain)

    chain = '*' + ''.join(
        self.parent_class_chains) + self._gen_class_chain()
    if DEBUG:
        print('CHAIN: %s', chain)
    return self._wdasearch_single('class chain', chain)

def find_element(self):
    """
    Returns:
        True, Element: single element
        False, error info
    """
    # element = None
    # elementId = self.find_element_id()
    isFound, respInfo = self.find_element_id()

    if DEBUG:
        # print('elementId=%s' % elementId)
        print('isFound=%s, respInfo=%s' % (isFound, respInfo))

    if isFound:
        elementInfo = respInfo
        elementId = elementInfo["id"]

        # add bounds support
        curBounds = None
        if "rect" in elementInfo:
            rectDict = elementInfo["rect"]
            x = rectDict["x"]
            y = rectDict["y"]
            width = rectDict["width"]
            height = rectDict["height"]
            curBounds = Rect(x, y, width, height)
        if DEBUG:
            print("curBounds=%s" % curBounds)

        # element = Element(self.http.new_client(''), elementId, rect=curRect)
        element = Element(self.http.new_client(''), elementId, bounds=curBounds)

        if DEBUG:
            print('element=%s' % element)

```

```

        respInfo = element

    return isFound, respInfo

def find_elements(self):
    """
    Returns:
        Element (list): all the elements
    """
    es = []
    for element_id in self.find_element_ids():
        if DEBUG:
            print('find_elements: element_id=%s' % element_id)
        ele = Element(self.http_new_client(''), element_id)
        if DEBUG:
            print('find_elements: ele=%s' % ele)
        es.append(ele)
    return es

def count(self):
    if DEBUG:
        print("count")

    elementIdList = self.find_element_ids()
    elementIdNum = len(elementIdList)

    if DEBUG:
        print("count: elementIdList=%s" % elementIdList)

    return elementIdNum

# def get(self, timeout=None, raise_error=True, isRespSingle=False):
# def get(self, timeout=None, raise_error=True, isRespSingle=True):
def get(self, timeout=None):
    """
    Args:
        timeout (float): timeout for query element, unit seconds
        Default 10s

    Returns:
        True, Element
        False, error info

    Raises:
    """
    isFound, respInfo = False, "Unknown error"

    if DEBUG:
        # print("get: timeout=%s, raise_error=%s, isRespSingle=%s" % (timeout, raise_error, isRespSingle))
        print("get: timeout=%s" % timeout)

```

```

start_time = time.time()
if timeout is None:
    timeout = self.timeout

if DEBUG:
    print("get: timeout=%s" % timeout)

endTime = start_time + timeout
while True:
    # if isRespSingle:
    #     singleElement = self.find_element()
    #     if singleElement:
    #         return singleElement
    # else:
    #     elems = self.find_elements()
    #     if len(elems) > 0:
    #         return elems[0]
    isFound, respInfo = self.find_element()
    if isFound:
        return isFound, respInfo

curTime = time.time()
if endTime < curTime:
    break

# time.sleep(0.01)
time.sleep(RETRY_INTERVAL)

if DEBUG:
    from datetime import datetime
    curDatetime = datetime.fromtimestamp(float(curTime))
    print("curDatetime=%s -> Not timeout, continue find" % curDatetime)

## check alert again
# sessionAlertExists = self.session.alert.exists
# httpAlertCallback = self.http.alert_callback

# if DEBUG:
#     print("get: sessionAlertExists=%s, httpAlertCallback=%s" % (sessionAlertExists, httpAlertCallback))

# if sessionAlertExists and httpAlertCallback:
#     self.http.alert_callback()

#     if DEBUG:
#         print("get: timeout=%s, raise_error=%s" % (timeout, raise_error))

#     # return self.get(timeout, raise_error)
#     return self.get(timeout)

# if raise_error:
#     raise WDAElementNotFoundError("element not found", "timeout %.1f" % timeout)
return isFound, respInfo

```

```

def __getattr__(self, oper):
    if DEBUG:
        print("oper=%s" % oper)

    # return getattr(self.get(), oper)
    isFound, respInfo = self.get()

    if DEBUG:
        print("isFound=%s, respInfo=%s" % (isFound, respInfo))

    if isFound:
        element = respInfo
        return getattr(element, oper)
    else:
        return None

def set_timeout(self, s):
    """
    Set element wait timeout
    """
    self.timeout = s
    return self

def __getitem__(self, index):
    self.index = index
    return self

def child(self, *args, **kwargs):
    chain = self._gen_class_chain()
    kwargs['parent_class_chains'] = self.parent_class_chains + [chain]
    return Selector(self.http, self.session, *args, **kwargs)

@property
def exists(self):
    return len(self.find_element_ids()) > self.index

def click_exists(self, timeout=0):
    """
    Wait element and perform click

    Args:
        timeout (float): timeout for wait

    Returns:
        bool: if successfully clicked
    """
    # e = self.get(timeout=timeout, raise_error=False)
    # if e is None:
    #     return False
    isFound, respInfo = self.get(timeout=timeout)
    if isFound:
        e = respInfo

```

```

        e.click()
        return True
    else:
        return False

# def wait(self, timeout=None, raise_error=True):
def wait(self, timeout=None):
    """ alias of get
    Args:
        timeout (float): timeout seconds

    Raises:
    """
    # return self.get(timeout=timeout, raise_error=raise_error)
    return self.get(timeout=timeout)

def wait_gone(self, timeout=None, raise_error=True):
    """
    Args:
        timeout (float): default timeout
        raise_error (bool): return bool or raise error

    Returns:
        bool: works when raise_error is False

    Raises:
        WDAElementNotDisappearError
    """
    start_time = time.time()
    if timeout is None or timeout <= 0:
        timeout = self.timeout
    while start_time + timeout > time.time():
        if not self.exists:
            return True
        if not raise_error:
            return False
        raise WDAElementNotDisappearError("element not gone")

    # todo
    # pinch
    # touchAndHold
    # dragfromtoforduration
    # twoFingerTap

    # todo
    # handleGetIsAccessibilityContainer
    # [[FBRoute GET:@"/wda/element/:uuid/accessibilityContainer"] respondWithTarget:self action:@selector(handleGetIsAccessibilityContainer:)],

class Element(object):
    # def __init__(self, httpclient, id):

```

```

# def __init__(self, httpclient, id, rect=None):
def __init__(self, httpclient, id, bounds=None):
    """
    base_url eg: http://localhost:8100/session/$SESSION_ID
    """
    self.http = httpclient
    self._id = id
    # add cached rect=bounds, to avoid later get rect but return 0
    if DEBUG:
        print("bounds=%s" % bounds)
    if bounds:
        self.bounds = bounds
        if DEBUG:
            print("use pass in bounds=%s" % self.bounds)

def __repr__(self):
    return '<wda.Element(id=%s)>' % self.id

def _req(self, method, url, data=None):
    return self.http.fetch(method, '/element/' + self.id + url, data)

def _wda_req(self, method, url, data=None):
    return self.http.fetch(method, '/wda/element/' + self.id + url, data)

def _prop(self, key):
    return self._req('get', '/' + key.lstrip('/')).value

def _wda_prop(self, key):
    ret = self._request('GET', 'wda/element/%s/%s' % (self._id, key)).value
    return ret

# @property
@cached_property
def id(self):
    return self._id

# @property
@cached_property
def label(self):
    return self._prop('attribute/label')

# @property
@cached_property
def className(self):
    return self._prop('attribute/type')

# @property
@cached_property
def text(self):
    return self._prop('text')

# @property
@cached_property

```

```

def name(self):
    # return self._prop('name')
    # Note: here property name, internally: FBElementCommands.m -> handleGetName -> use wdType
    # so, for:
    # <XCUIElementTypeButton type="XCUIElementTypeButton" name="¥1.00" label="¥1.00" enabled="true" visible="true" x="154" y="152" width="74" height="30"/>
    # name is XCUIElementTypeButton in type="XCUIElementTypeButton"
    # not expected: ¥1.00 in name="¥1.00"
    # so change to 'attribute/name'
    return self._prop('attribute/name') # ¥1.00

# @property
@cached_property
def displayed(self):
    return self._prop("displayed")

# @property
@cached_property
def enabled(self):
    return self._prop('enabled')

# @property
@cached_property
def accessible(self):
    return self._wda_prop("accessible")
    # return self._prop("accessible")

# @property
@cached_property
def accessibility_container(self):
    return self._wda_prop('accessibilityContainer')
    # return self._prop('accessibilityContainer')

@property
# Special: not used cache value for AppStore downloading element value will changed
#   <XCUIElementTypeButton type="XCUIElementTypeButton" value="18%" name="正在下载" label="正在下载" enabled="true" visible="true" x="154" y="308" width="74" height="30"/>
    # if use cache, later always get None
    # @cached_property
def value(self):
    curValue = self._prop('attribute/value')
    # curValue = self._prop('attribute/wdValue')
    if DEBUG:
        print("curValue=%s" % curValue)
    return curValue

# @property
@cached_property
def enabled(self):
    return self._prop('enabled')

# @property

```

```

@cached_property
def visible(self):
    # if DEBUG:
    #     print("call attribute/visible to get visible value")
    return self._prop('attribute/visible')

# @property
@cached_property
def bounds(self):
    if DEBUG:
        print("try get Element bounds=rect")

    value = self._prop('rect')
    x, y = value['x'], value['y']
    w, h = value['width'], value['height']
    rectObj = Rect(x, y, w, h)
    if DEBUG:
        print("rectObj=%s" % rectObj)
    return rectObj

# operations
def tap(self):
    return self._req('post', '/click')

def click(self):
    """ Alias of tap """
    return self.tap()

def tap_hold(self, duration=1.0):
    """
    Tap and hold for a moment

    Args:
        duration (float): seconds of hold time

    [[FBRoute POST:@"/wda/element/:uuid/touchAndHold"] respondWithTarget:self action:@
    selector(handleTouchAndHold:)],
    """
    return self._wda_req('post', '/touchAndHold', {'duration': duration})

def scroll(self, direction='visible', distance=1.0):
    """
    Args:
        direction (str): one of "visible", "up", "down", "left", "right"
        distance (float): swipe distance, only works when direction is not "visible"

    Raises:
        ValueError

    distance=1.0 means, element (width or height) multiply 1.0
    """
    if direction == 'visible':
        self._wda_req('post', '/scroll', {'toVisible': True})

```

```
        elif direction in ['up', 'down', 'left', 'right']:
            self._wda_req('post', '/scroll', {
                'direction': direction,
                'distance': distance
            })
        else:
            raise ValueError("Invalid direction")
    return self

    def pinch(self, scale, velocity):
        """
        Args:
            scale (float): scale must > 0
            velocity (float): velocity must be less than zero when scale is less than 1

        Example:
            pinchIn -> scale:0.5, velocity: -1
            pinchOut -> scale:2.0, velocity: 1
        """
        data = {'scale': scale, 'velocity': velocity}
        return self._wda_req('post', '/pinch', data)

    def set_text(self, value):
        return self._req('post', '/value', {'value': value})

    def clear_text(self):
        return self._req('post', '/clear')

# def child(self, **kwargs):
#     return Selector(self.__base_url, self._id, **kwargs)

# todo lot of other operations
# tap_hold
```



crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2021-04-11 11:35:48

元素处理

此处整理用wda查找元素、点击元素等常见代码段。

getiOSElementQuery 生成wda的元素的query

```

def getiOSElementQuery(self, locator):
    """from element locator to generate iOS element query"""
    query = None
    # if isinstance(locator, list):
    #     logging.error("TODO: add support in future for locator is list in getiOSElementQuery")
    # elif isinstance(locator, dict):
    #     {'name': '公众号'}
    #     query = {}
    #     query = locator
    query = copy.deepcopy(locator)

    # Note: not auto convert key for compatible old Android key
    # {'text': '通讯录'}, {'desc': '添加'}
    # locatorText = locator.get("text")
    # locatorDesc = locator.get("desc")
    # if locatorText:
    #     query = {"name": locatorText}
    # elif locatorDesc:
    #     query = {"label": locatorDesc}
    # else:
    #     logging.warning("TODO: add support later for %s", locator)

    # auto add type for iOS
    hasTag = "tag" in query.keys()
    noType = "type" not in query.keys()
    if hasTag and noType:
        query["type"] = query["tag"]
        del query["tag"]

    # 20200402: facebook-wda+WebDriverAgent internally
    #     (1) not support visible
    #     (2) sometime support accessible
    # -> now, not use visible anymore
    # 20200430: updated latest WebDriverAgent source code, now has supported visible
    #     so above not need delete visible
    # 20200430: but after add visible, still not find (some) element, so remove again
    if "visible" in query.keys():
        del query["visible"]

    logging.debug("locator=%s -> query=%s", locator, query)
    return query

```

说明：

对于输入的query的dict，进行一定处理：

1. 去掉visible
 - 截止20200615， WebDriverAgent内部还是不能完美支持元素的visible值的判断
 - -» 有时候支持 有时候不支持， 导致传入visible=true， 却找不到元素
 - -» 所以去掉visible
2. 把tag改为type

调用举例：

```
def isFoundElement_iOS(self, locator):
    isFound = False
    foundElement = None
    query = self.getIOSElementQuery(locator)
    if query:
        isFound, foundElement = self.findElement(query, query)
    return isFound, foundElement
```

findElement 查找元素

```
# def findElement(self, query={}, timeout=WaitFind):
# def findElement(self, query={}, timeout=WaitFind, isRetryNoYWhenFail=False):
def findElement(self, query {}, timeout WaitFind, isRetryNoYWhenFail True):
    """Find element from query

Args:
    query (dict): query condition
    timeout(float): max wait time
    isRetryNoYWhenFail(bool): True to do retry but remove y from query when found fail
    , False to not retry
Returns:
    bool, Element/str
    True, Element
    False, error message
Raises:
    """
    logging.debug("query=%s", query)
    isFound, respInfo = False, "Unknown error"

    elementSelector = self.curSession(**query, timeout timeout)
    logging.debug("elementSelector=%s", elementSelector)
    isFound, respInfo = elementSelector.get()
    logging.debug("query=%s -> isFound=%s, respInfo=%s", query, isFound, respInfo)
    if not isFound:
        if isRetryNoYWhenFail and ("y" in query):
            # Note:
            # for sometime / many cases:
            #     not found element due to y position is not correct but other property(x,
```

```

width, height, name, ...) all correct
    # for these cases, retry to find without y
    noYQuery = copy.deepcopy(query)
    del noYQuery["y"]
    elementSelectorNoY = self.curSession(noYQuery, timeout=timeout)
    logging.debug("elementSelectorNoY=%s", elementSelectorNoY)
    isFound, respInfo = elementSelectorNoY.get()
    logging.debug("retry no y: noYQuery=%s -> isFound=%s, respInfo=%s", noYQuery,
    isFound, respInfo)
    # sometime here found out element but y is negative:
    # <XCUIElementTypeImage type="XCUIElementTypeImage" name="main_avatar" enabled="true" visible="false" x="16" y="-146" width="35" height="36"/>
    # actually is not visible -> not our expected -> should filter out, consider a
    s not found
    if isFound:
        curRect = respInfo.bounds
        logging.debug("curRect=%s", curRect)
        isValidY = curRect.y >= 0
        isVisible = respInfo.visible
        logging.debug("isVisible=%s", isVisible)
        isReallyFound = isValidY and isVisible
        logging.debug("isReallyFound=%s", isReallyFound)
        if not isReallyFound:
            isFound = False
            respInfo = respInfo

    return isFound, respInfo

```

调用举例：

- (1) 简单的
- (2) 复杂的，带parent的class chain：设置 WiFi列表页 中查找 无线局域网

对于xml

```

<XCUIElementTypeNavigationBar type="XCUIElementTypeNavigationBar" name="无线局域网" enabled=
"true" visible="true" x="0" y="20" width="414" height="44">
    <XCUIElementTypeButton type="XCUIElementTypeButton" name="设置" label="设置" enabled="t
rue" visible="true" x="0" y="20" width="66" height="44"/>
    <XCUIElementTypeOther type="XCUIElementTypeOther" name="无线局域网" label="无线局域网" e
nabled="true" visible="true" x="165" y="31" width="88" height="21"/>
</XCUIElementTypeNavigationBar>

```

写法是：

```

wifiName = "无线局域网"
parentNaviBarClassChain = "/XCUIElementTypeNavigationBar[`name = '%s' AND rect.x = 0 AND r
ect.width = %d`]" % (wifiName, self.X)
wifiQuery = {"type": "XCUIElementTypeOther", "name": wifiName, "enabled": "true"}
wifiQuery["parent_class_chains"] = [parentNaviBarClassChain]

```

```
isFoundWifi, respInfo = self.findElement(query wifiQuery, timeout 0.1)
```

和：

```
<XCUIElementTypeTabBar type="XCUIElementTypeTabBar" enabled="true" visible="true" x="0" y="680" width="414" height="56">
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="680" width="414" height="56">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="680" width="414" height="56">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="680" width="414" height="56"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="680" width="414" height="56"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="680" width="414" height="56"/>
        </XCUIElementTypeOther>
    </XCUIElementTypeOther>
    <XCUIElementTypeButton type="XCUIElementTypeButton" value="2" name="微信" label="微信" enabled="true" visible="true" x="2" y="681" width="100" height="55"/>
    <XCUIElementTypeButton type="XCUIElementTypeButton" name="通讯录" label="通讯录" enabled="true" visible="true" x="106" y="681" width="99" height="55"/>
    <XCUIElementTypeButton type="XCUIElementTypeButton" value="未读" name="发现" label="发现" enabled="true" visible="true" x="209" y="681" width="100" height="55"/>
    <XCUIElementTypeButton type="XCUIElementTypeButton" name="我" label="我" enabled="true" visible="true" x="313" y="681" width="99" height="55"/>
</XCUIElementTypeTabBar>
```

代码 判断iOS中页面是否处于 微信：

```
weixinTabQuery = {"type": "XCUIElementTypeButton", "name": "微信", "label": "微信", "enabled": "true"}
weixinTabQuery["parent_class_chains"] = [ tabBarClassChain ]
isFoundWeixin, respInfo = self.findElement(query weixinTabQuery, timeout 0.2)
iOSisInWeixin = isFoundWeixin
return iOSisInWeixin
```

(3) 复杂的， name中包含的

```
<XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="↓下拉返回商品详情" name="↓下拉返回商品详情" label="↓下拉返回商品详情" enabled="true" visible="false" x="0" y="9" width="414" height="54"/>
```

写法：

```
DropDownReturnDetailStr = "下拉返回商品详情"
dropDownReturnDetailQuery = {"type": "XCUIElementTypeStaticText", "nameContains": DropDownR}
```

```

    returnDetailStr)
    isFound, foundElement = self.findElement(dropDownReturnDetailQuery, timeout 0.2)

```

- AppStore 详情页 不折叠输入法 带金额的购买按钮 ¥1.00:

```

<XCUIElementTypeCollectionView type="XCUIElementTypeCollectionView" enabled="true" visible=
"x=true" x=="0" y=="0" width="414" height="736">
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="0"
y="-672" width="414" height="736"/>
    <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="0" y=
64" width="414" height="211">
        <XCUIElementTypeImage type="XCUIElementTypeImage" value="不折叠输入法" name="插图" 1
abel="插图" enabled="true" visible="true" x="20" y="64" width="118" height="118"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="不折叠输入法" name=
"不折叠输入法" label="不折叠输入法" enabled="true" visible="true" x="154" y="71" width="134"
height="27"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="JUNJIE HUANG" n
ame="JUNJIE HUANG" label="JUNJIE HUANG" enabled="true" visible="false" x="154" y="101" wid
th="108" height="19"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="新版发文不再一行
的小键盘" name="新版发文不再一行的小键盘" label="新版发文不再一行的小键盘" enabled="true" visible
="true" x="154" y="101" width="184" height="19"/>
        <XCUIElementTypeButton type="XCUIElementTypeButton" name="¥1.00" label="¥1.00" en
abled="true" visible="true" x="154" y="152" width="74" height="30"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="App 内购买项目" n
ame="App 内购买项目" label="App 内购买项目" enabled="true" visible="true" x="234" y="152" w
idth="62" height="30"/>
        <XCUIElementTypeButton type="XCUIElementTypeButton" name="更多" label="更多" enabled
="true" visible="true" x="366" y="154" width="28" height="28"/>
        <XCUIElementTypeOther type="XCUIElementTypeOther" name="三颗星, 793个评分, 3, 工具,
4+, 年龄" label="三颗星, 793个评分, 3, 工具, 4+, 年龄" enabled="true" visible="true" x="20" y=
"202" width="374" height="50"/>
    </XCUIElementTypeCell>

```

查找写法是：

```

parentCellClassChain = "/XCUIElementTypeCell[`rect.x = 0 AND rect.width = %d`]" % self.X
moneyButtonQuery = {"type": "XCUIElementTypeButton", "nameContains": "¥", "enabled": "true"}

moneyButtonQuery["parent_class_chains"] = [ parentCellClassChain ]
foundMoneyButton, respInfo = self.findElement(query=moneyButtonQuery, timeout 0.1)

```

(4) 复杂的, name符合条件的

场景：寻找 微信公众号列表页中 上下滚动后的 顶部浮动的大写字母横条的元素

```

<XCUIElementTypeOther type="XCUIElementTypeOther" name="D" enabled="true" visible="true" x=
"0" y="64" width="375" height="22">
    <XCUIElementTypeOther type="XCUIElementTypeOther" name="D" label="D" enabled="true" vi

```

```

    sible="true" x="10" y="64" width="12" height="22"/>
</XCUIElementTypeOther>

<XCUIElementTypeOther type="XCUIElementTypeOther" name="E" enabled="true" visible="true" x=
"0" y="64" width="375" height="22">
    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="E" name="E" label="E"
" enabled="true" visible="true" x="10" y="64" width="10" height="22"/>
</XCUIElementTypeOther>

<XCUIElementTypeOther type="XCUIElementTypeOther" name="J" enabled="true" visible="true" x=
"0" y="64" width="375" height="22">
    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="J" name="J" label="J"
" enabled="true" visible="true" x="10" y="64" width="9" height="22"/>
</XCUIElementTypeOther>

```

代码：

```

floatUpperLetterElement = None
# floatUpperLetterQuery = {"type": "XCUIElementTypeStaticText", "valueMatches": "^[A-Z]$"}
# isFound, foundElement = findElement(curSession, floatUpperLetterQuery)
floatUpperLetterQuery = {"type": "XCUIElementTypeOther", "nameMatches": "^[A-Z]$", "enabled":
"true", "x": "0", "width": "%s" % self.X}
isFound, foundElement = self.findElement(floatUpperLetterQuery)

```

(5) 复杂的，value包含的

```

curNoticeQuery = {"type": "XCUIElementTypeStaticText", "valueContains": curStepNotice, "en
bled": "true"}
curNoticeQuery["parent_class_chains"] = [scrollViewClassChain]
isFoundCurNotice, respInfo = self.findElement(query=curNoticeQuery, timeout 0.5)

```

findAndClickElement 查找并点击元素

```

# def findAndClickElement(self, query={}, timeout=WaitFind, enableDebug=False):
# def findAndClickElement(self, query={}, timeout=WaitFind, isShowErrLog=True):
# def findAndClickElement(self, query={}, timeout=WaitFind, isShowErrLog=True, isRetryNoW
henFail=True):
def findAndClickElement(self, query {}, timeout WaitFind, isShowErrLog True):
    """Find and click element

Args:
    query (dict): query parameter for find element
    timeout (float): max timeout seconds for find element
    enableDebug (bool): if enable debug, then draw clicked rectange for current screen
    isShowErrLog(bool): True to show error log, False to not show logging.error when e
rror
Returns:
    bool

```

```

Raises:
"""
foundAnClicked = False

# isFound, respInfo = self.findElement(query=query, timeout=timeout)
# isFound, respInfo = self.findElement(query=query, timeout=timeout, isRetryNoYWhenFail=isRetryNoYWhenFail)
    isFound, respInfo = self.findElement(query=query, timeout=timeout)
logging.debug("isFound=%s, respInfo=%s", isFound, respInfo)
if isFound:
    curElement = respInfo

    # if enableDebug:
    #     # for debug
    #     curScreenFile = debugSaveScreenshot(curScale=self.curSession.scale)
    #     utils.imageDrawRectangle(curScreenFile, rectLocation=curElement.bounds)

    clickOk = self.clickElement(curElement)
    logging.info("%s to Clicked element %s", clickOk, query)

    foundAnClicked = clickOk
else:
    if isShowErrLog:
        logging.error("Not found element %s", query)
    else:
        logging.debug("Not found element %s", query)

return foundAnClicked

```

调用举例:

(1)

```

# <XCUIElementTypeButton type="XCUIElementTypeButton" name="Search" label="Search" enabled="true" visible="true" x="281" y="620" width="94" height="47"/>
# <XCUIElementTypeButton type="XCUIElementTypeButton" name="Search" label="搜索" enabled="true" visible="true" x="309" y="685" width="103" height="50"/>
# searchButtonQuery = {"name": "Search"}
searchButtonQuery = {"name": "Search", "type": "XCUIElementTypeButton"}

foundAndClickedDoSearch = self.findAndClickElement(searchButtonQuery, timeout 1.0)

```

(2) 场景：设置 WiFi 配置代理 从手动切换到 关闭 存储

xml:

```

<XCUIElementTypeNavigationBar type="XCUIElementTypeNavigationBar" name="配置代理" enabled="true" visible="true" x="0" y="20" width="414" height="44">
    <XCUIElementTypeButton type="XCUIElementTypeButton" name="xxx_guest_5G" label="xxx_guest_5G" enabled="true" visible="true" x="0" y="20" width="155" height="44"/>
    <XCUIElementTypeOther type="XCUIElementTypeOther" name="配置代理" label="配置代理" enabled="true" visible="true" x="172" y="31" width="70" height="21"/>

```

```

<XCUIElementTypeButton type="XCUIElementTypeButton" name="存储" label="存储" enabled="true" visible="true" x="359" y="20" width="43" height="44"/>
</XCUIElementTypeNavigationBar>

```

代码：

```

storeName = "存储"
parentNaviBarClassChain = "/XCUIElementTypeNavigationBar['name = '配置代理' AND rect.x = 0 AND rect.width = %d']" % self.X
storeButtonQuery = {"type": "XCUIElementTypeButton", "name": storeName, "enabled": "true"}
storeButtonQuery["parent_class_chains"] = [parentNaviBarClassChain]
foundAndClickedStore = self.findAndClickElement(query=storeButtonQuery, timeout=0.1)

```

(3) isShowErrLog=False 当找不到，不要报错

场景：

AppStore 详情页 淘宝 弹框 安装

xml

```

<XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="515" width="414" height="221">
    <XCUIElementTypeTable type="XCUIElementTypeTable" enabled="true" visible="true" x="0" y="515" width="414" height="120">
        <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="0" y="515" width="414" height="77">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="16" y="590" width="398" height="1"/>
            <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="false" x="55" y="533" width="40" height="41"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="手机淘宝 - 淘到你说好 TAOBAO (CHINA) SOFTWARE CO.,LTD APP" name="手机淘宝 - 淘到你说好 TAOBAO (CHINA) SOFTWARE CO.,LTD APP" label="手机淘宝 - 淘到你说好 TAOBAO (CHINA) SOFTWARE CO.,LTD APP" enabled="true" visible="true" x="110" y="530" width="234" height="48"/>
        </XCUIElementTypeCell>
        <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="0" y="591" width="414" height="45">
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="账户" name="账户" label="账户" enabled="true" visible="true" x="16" y="606" width="79" height="16"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="xxx@xxx.com" name="xxx@xxx.com" label="xxx@xxx.com" enabled="true" visible="true" x="111" y="606" width="134" height="16"/>
        </XCUIElementTypeCell>
    </XCUIElementTypeTable>
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="635" width="414" height="101">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="16" y="635" width="398" height="1"/>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="655" width="414" height="69"/>
        <XCUIElementTypeButton type="XCUIElementTypeButton" name="安装" label="安装" enabled="true" visible="true" x="110" y="655" width="234" height="48"/>
    </XCUIElementTypeOther>

```

```
= "true" visible="true" x="165" y="667" width="84" height="37"/>
</XCUIElementTypeOther>
</XCUIElementTypeOther>
```

代码：

```
parentOtherClassChain = "/XCUIElementTypeOther[`rect.x = 0 AND rect.width = %d`]" % self.x
installButtonQuery = {"type": "XCUIElementTypeButton", "name": "安装", "enabled": "true"}
installButtonQuery["parent_class_chains"] = [parentOtherClassChain]
foundAndClickedInstall = self.findAndClickElement(query=installButtonQuery, timeout=0.1, i
sShowErrLog=False)
```

findAndClickButtonElementBySoup

```
def findAndClickButtonElementBySoup(self, curButtonSoup=None, curButtonName=None):
    """
        iOS的bug：根据bs找到了soup元素（往往是一个button）后，用 clickCenterPosition=click
        ElementCenterPosition 去点击中间坐标，往往会有问题
        实际上点击的是别的位置，别的元素
        为了规避此bug，所以去：
        通过soup，再去找button的wda的元素，然后根据元素去点击
        则都是可以正常点击，不会有误点击的问题
    """

    # # change to wda element query then click by element
    # if not curButtonName:
    #     curSoupAttrs = curButtonSoup.attrs
    #     curButtonName = curSoupAttrs["name"]
    # # rights close white
    # # login close
    # curButtonQuery = {"type": "XCUIElementTypeButton", "enabled": "true", "name": curB
    uttonName}
    # # foundAndClicked = self.findAndClickElement(curButtonQuery)

    curButtonQuery = {"type": "XCUIElementTypeButton", "enabled": "true"}
    extraQuery = {}

    # change to wda element query then click by element
    if curButtonName:
        extraQuery["name"] = curButtonName
    else:
        if curButtonSoup:
            curSoupAttrs = curButtonSoup.attrs
            if hasattr(curSoupAttrs, "name"):
                curButtonName = curSoupAttrs["name"]
            # rights close white
            # login close
            extraQuery["name"] = curButtonName
        else:
            # no name attribute, use position
```

```

        x = curSoupAttrs["x"]
        y = curSoupAttrs["y"]
        width = curSoupAttrs["width"]
        height = curSoupAttrs["height"]
        extraQuery["x"] = x
        extraQuery["y"] = y
        extraQuery["width"] = width
        extraQuery["height"] = height
        # {'enabled': 'true', 'height': '32', 'type': 'XCUIElementTypeButton',
        'width': '31', 'x': '339', 'y': '122'}

        # merge query
        # curButtonQuery = {**curButtonQuery, **extraQuery}
        curButtonQuery.update(extraQuery) # {'enabled': 'true', 'height': '32', 'type': 'XCUIElementTypeButton', 'width': '32', 'x': '338', 'y': '150'}

    foundAndClicked = self.findAndClickElement(curButtonQuery, isShowErrLog False)
    return foundAndClicked

```

不同的地方的各种调用：

```

foundAndProcessedPopup = self.findAndClickButtonElementBySoup(possibleCloseSoup)

foundAndProcessedPopup = self.findAndClickButtonElementBySoup(commonCloseSoup)

foundAndProcessedPopup = self.findAndClickButtonElementBySoup(wifiCellularSoup)

foundAndProcessedPopup = self.findAndClickButtonElementBySoup(okSoup)

foundAndProcessedPopup = self.findAndClickButtonElementBySoup(allowSoup)

foundAndProcessedPopup = self.findAndClickButtonElementBySoup(curButtonName="取消")

```

详见：

【已解决】自动抓包iOS的app：优化clickElementCenterPosition点击失效时换用wda寻找元素并点击逻辑

findAndClickCenterPosition 查找并点击元素中间位置

```

def findAndClickCenterPosition(self, bsChainList, soup None, isUseWdaQueryAndClick False):
    """use BeautifulSoup chain list to find soup node then click node center position

    Args:
        bsChainList (list): dict list for dict of tag and attrs
        soup (Soup)): BeautifulSoup soup
        isUseWdaQueryAndClick (bool): for special node bs click not work, so need change to wda query element then click by element
    Returns:
        bool: found and clicked or not

```

```

Raises:
"""
foundAndClicked = False
if not soup:
    curPageXml = self.get_page_source()
    soup = CommonUtils.xmlToSoup(curPageXml)
foundSoup = CommonUtils.bsChainFind(soup, bsChainList)
if foundSoup:
    if isUseWdaQueryAndClick:
        foundAndClicked = self.findAndClickButtonElementBySoup(foundSoup)
    else:
        self.clickElementCenterPosition(foundSoup)
        foundAndClicked = True

return foundAndClicked

```

场景：米家 弹框 立即体验 xml

```

<XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0"
width="414" height="736">
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0"
width="414" height="736">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0"
width="414" height="736">
            <XCUIElementTypeImage type="XCUIElementTypeImage" name="sort_guide_icon" enabled="true"
visible="false" x="54" y="115" width="306" height="310"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="设备排序全新
升级" name="设备排序全新升级" label="设备排序全新升级" enabled="true" visible="true" x="111" y="426"
width="192" height="34"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="1.新增快速置
顶，排序操作更方便 2.顶部排序固定，不会被随意打乱" name="1.新增快速置顶，排序操作更方便 2.顶部排序
固定，不会被随意打乱" label="1.新增快速置顶，排序操作更方便 2.顶部排序固定，不会被随意打乱" enabled="true"
visible="true" x="95" y="468" width="224" height="42"/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="立即体验" label="立即
体验" enabled="true" visible="true" x="24" y="657" width="366" height="43"/>
        </XCUIElementTypeOther>
    </XCUIElementTypeOther>
</XCUIElementTypeOther>

```

调用举例：

```

immediatelyExperienceChainList = [
    {
        "tag": "XCUIElementTypeOther",
        "attrs": self.FullScreenAttrDict
    },
    {
        "tag": "XCUIElementTypeOther",
        "attrs": self.FullScreenAttrDict
    },
    {

```

```

        "tag": "XCUIElementTypeButton",
        "attrs": {"enabled":"true", "visible":"true", "name":"立即体验"}
    },
]
foundAndProcessedPopup = self.findAndClickCenterPosition(immediatelyExperienceChainList, s
oup)
return foundAndProcessedPopup

```

`findAndClickButtonElementBySoup` 通过soup去用wda的query查找元素并点击

```

def findAndClickButtonElementBySoup(self, curButtonSoup=None, curButtonName=None):
    """
        iOS的bug: 根据bs找到了soup元素(往往是一个button)后, 用 clickCenterPosition=clickElem
        entCenterPosition 去点击中间坐标, 往往会有问题
        实际上点击的是别的位置, 别的元素
        为了规避此bug, 所以去:
        通过soup, 再去找button的wda的元素, 然后根据元素去点击
        则都是可以正常点击, 不会有误点击的问题
    """

    # # change to wda element query then click by element
    # if not curButtonName:
    #     curSoupAttrs = curButtonSoup.attrs
    #     curButtonName = curSoupAttrs["name"]
    # # rights close white
    # # login close
    # curButtonQuery = {"type": "XCUIElementTypeButton", "enabled": "true", "name": curButto
    nName}
    # # foundAndClicked = self.findAndClickElement(curButtonQuery)

    curButtonQuery = {"type": "XCUIElementTypeButton", "enabled": "true"}
    extraQuery = {}

    # change to wda element query then click by element
    if curButtonName:
        extraQuery["name"] = curButtonName
    else:
        if curButtonSoup:
            curSoupAttrs = curButtonSoup.attrs
            if hasattr(curSoupAttrs, "name"):
                curButtonName = curSoupAttrs["name"]
            # rights close white
            # login close
            extraQuery["name"] = curButtonName
        else:
            # no name attribute, use position
            x = curSoupAttrs["x"]
            y = curSoupAttrs["y"]
            width = curSoupAttrs["width"]
            height = curSoupAttrs["height"]
            extraQuery["x"] = x

```

```

        extraQuery["y"] = y
        extraQuery["width"] = width
        extraQuery["height"] = height
        # {'enabled': 'true', 'height': '32', 'type': 'XCUIElementTypeButton', 'width': '31', 'x': '339', 'y': '122'}

    # merge query
    # curButtonQuery = {**curButtonQuery, **extraQuery}
    curButtonQuery.update(extraQuery) # {'enabled': 'true', 'height': '32', 'type': 'XCUIElementTypeButton', 'width': '32', 'x': '338', 'y': '150'}

foundAndClicked = self.findAndClickElement(curButtonQuery, isShowErrLog=False)
return foundAndClicked

```

说明：

- iOS的bug：根据bs找到了soup元素（往往是一个button）后，用clickElementCenterPosition=clickElementCenterPosition去点击中间坐标，往往会有问题
 - 实际上点击的是别的位置，别的元素
- 为了规避此bug，所以去：
 - 通过soup，再去找button的wda的元素，然后根据元素去点击
 - 则都是可以正常点击，不会有误点击的问题

调用举例：

```

commonCloseSoup = CommonUtils.bsChainFind(soup, commonCloseChainList)
if commonCloseSoup:
    # self.clickElementCenterPosition(commonCloseSoup) # sometime not work
    # foundAndProcessedPopup = True

    # so change to wda query element then click
    foundAndProcessedPopup = self.findAndClickButtonElementBySoup(commonCloseSoup)

```

`clickElementCenterPosition` 点击元素（通过属性中找到元素坐标，点击中间位置）

```

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

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

```

```

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

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

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

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

return hasClicked

```

调用举例：

```

confirmSoup = parentOtherSoup.find(
    "XCUIElementTypeButton",
    attrs={"visible": "true", "enabled": "true", "name": "确定"})
)
if confirmSoup:
    self.clickElementCenterPosition(confirmSoup)
    foundAndProcessedPopup = True

```

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09 10:13:38

iOS设备

此处整理和iOS设备，比如iPhone相关的常用代码段。

设备是否解锁

```
def is_device_unlock_iOS(self, device):
    # Note: before later real init dirver, init here for here can use wdaClient
    self.init_device_driver_iOS()
    isLocked = self.wdaClient.locked()
    isUnlocked = not isLocked
    return isUnlocked
```

注：其中 `init_device_driver_iOS` 定义详见：[wda基本的初始化](#)

获取电池状态

```
def is_device_charged_iOS(self):
    batteryInfo = self.wdaClient.battery_info()
    logging.debug("batteryInfo=%s", batteryInfo)
    # batteryInfo={'level': 1, 'state': 2}
    # batteryInfo={'level': 0.49000000953674316, 'state': 2}
    batteryLevelFloat = batteryInfo["level"]
    batteryLevelPercentInt = int(batteryLevelFloat * 100) # 100, 49
    batteryStateInt = batteryInfo["state"]
    logging.debug("batteryStateInt=%s", batteryStateInt)

    curBattrystate = BatteryState(batteryStateInt)
    logging.debug("curBattrystate=%s", curBattrystate) # curBattrystate=BatteryState.Charging
    curBattrystateName = curBattrystate.name
    logging.debug("curBattrystateName=%s", curBattrystateName) # curBattrystateName=Charging
    logging.info("Battery state: %s", curBattrystateName) # Battery state: Charging
    return batteryLevelPercentInt
```

相关定义：

```
from wda import ScreenshotQuality, BatteryState, ApplicationState
```

其中是我自己定义的：

```
#####
# Definition
#####
```

```
# https://developer.apple.com/documentation/uikit/uidevice/1620042-batterylevel?language=objc
class BatteryState(Enum):
    Unknown = 0
    Unplugged = 1
    Charging = 2
    Full = 3

# https://developer.apple.com/documentation/xctest/xctimagequality?language=objc
class ScreenshotQuality(Enum):
    Original = 0 # lossless PNG image
    Medium = 1 # high quality lossy JPEG image
    Low = 2 # highly compressed lossy JPEG image

# https://developer.apple.com/documentation/xctest/xcuiapplicationstate?language=objc
class ApplicationState(Enum):
    Unknown = 0
    NotRunning = 1
    RunningBackgroundSuspended = 2
    RunningBackground = 3
    RunningForeground = 4
```

get_devices_iOS 当前连接（到Mac）的iOS设备

```
def get_devices_iOS(self):
    deviceStrList = self.get_iOS_deviceStrList()
    iOSDevIdList = []
    for eachDevStr in deviceStrList:
        # Mo iPhone (11.4.1) [fc9e1f9f2e1339328b9580f826a30fded3bc69ff]
        # iPhone 11 Pro Max (13.3) + Apple Watch Series 5 - 44mm (6.1.1) [D86F0BD5-4D38-45
37-9C8C-2F5C74E404CA] (Simulator)
        # iPhone 11 (13.3) [509BC7C7-9C0E-42FA-8AB2-F5220EBAA13B] (Simulator)
        foundDevId = re.search("\[(?P<iOSdevId>[\w-]+)\](\s+\\(Simulator\\))?", eachDevStr,
re.I)
        if foundDevId:
            iOSdevId = foundDevId.group("iOSdevId")
            iOSDevIdList.append(iOSdevId)
    # ['fc9e1f9f2e1339328b9580f826a30fded3bc69ff', 'D86F0BD5-4D38-4537-9C8C-2F5C74E404CA',
...]
    return iOSDevIdList
```

相关函数：

```
def get_iOS_deviceStrList(self):
    deviceStrList = []
    deviceListCmd = 'instruments -s devices'
    deviceListStr = CommonUtils.get_cmd_lines(deviceListCmd, text=True)
    if deviceListStr:
```

```

deviceRawList = deviceListStr.split("\n")
"""

Known Devices:
limao的MacBook Pro [F9089371-1060-5CE3-99BB-81741693BE80]
Mo iPhone (11.4.1) [fc9e1f9f2e1339328b9580f826a30fded3bc69ff]
Apple TV (13.3) [6680F059-4DE1-430C-B696-228AC27CAA88] (Simulator)
Apple TV 4K (13.3) [048E58E8-6A27-4D81-BDEB-8812C610B756] (Simulator)
Apple TV 4K (at 1080p) (13.3) [384D5E60-B6B1-481E-BDC3-B7FF8F773412] (Simulator)
)
Apple Watch Series 4 - 40mm (6.1.1) [1B98415B-3FDE-401B-A80C-A3551DB207D7] (Simulator)
Apple Watch Series 4 - 44mm (6.1.1) [661838E9-B0BE-42B4-B55E-9A34263B1AEA] (Simulator)
iPad (7th generation) (13.3) [7F8EDE89-74E0-4BAB-B3CA-09E2DAE1F095] (Simulator)
)
iPad Air (3rd generation) (13.3) [BBC48526-3922-4C97-BA14-B1888385243A] (Simulator)
iPad Pro (11-inch) (13.3) [04DD3B8A-5B78-48E8-8B22-56796A9CFB73] (Simulator)
iPad Pro (12.9-inch) (3rd generation) (13.3) [D811684E-2F3E-4FC6-92EA-39301451F7E5] (Simulator)
iPad Pro (9.7-inch) (13.3) [B11D5D40-FEA2-4114-B053-E4CFD29D127C] (Simulator)
iPhone 11 (13.3) [509BC7C7-9C0E-42FA-8AB2-F5220EBAA13B] (Simulator)
iPhone 11 Pro (13.3) [3E8E7E92-66F2-4AF3-A405-23B5FB231DE7] (Simulator)
iPhone 11 Pro (13.3) + Apple Watch Series 5 - 40mm (6.1.1) [F76D77EF-0932-4164-94BB-9FC757420911] (Simulator)
iPhone 11 Pro Max (13.3) [50C15135-1532-44C5-B82C-B327F88F2712] (Simulator)
iPhone 11 Pro Max (13.3) + Apple Watch Series 5 - 44mm (6.1.1) [D86F0BD5-4D38-4537-9C8C-2F5C74E404CA] (Simulator)
iPhone 8 (13.3) [54589698-0C9F-407D-B21A-83432CABB681] (Simulator)
iPhone 8 Plus (13.3) [509B7103-97DB-4AB9-B829-001190ED4B7E] (Simulator)
"""

# CoreData: annotation: Failed to load optimized model at path '/Applications/Xcode.app/Contents/Applications/Instruments.app/Contents/Frameworks/InstrumentsPackaging.framework/Versions/A/Resources/XRPackageModel.momd/XRPackageModel 9.0.omo'
# InvalidCoreData = "CoreData"
# Known Devices:
# InvalidKnownDevices = "Known Devices"
for eachDeviceStr in deviceRawList:
    eachDeviceStr = eachDeviceStr.strip()
    # isNotCoreData = InvalidCoreData not in eachDeviceStr
    # isNotKnownDevices = InvalidKnownDevices not in eachDeviceStr
    # if isNotCoreData and isNotKnownDevices:
    # if isNotKnownDevices:
    # Note: current only support iPhone devices, both real and simulator
    #     with device name contain 'iphone'
    foundiPhone = re.search("iPhone", eachDeviceStr, re.I)
    if foundiPhone:
        deviceStrList.append(eachDeviceStr)
    """

    Mo iPhone (11.4.1) [fc9e1f9f2e1339328b9580f826a30fded3bc69ff]
    iPhone 11 (13.3) [509BC7C7-9C0E-42FA-8AB2-F5220EBAA13B] (Simulator)
    iPhone 11 Pro (13.3) [3E8E7E92-66F2-4AF3-A405-23B5FB231DE7] (Simulator)
)

```

```
        iPhone 11 Pro (13.3) + Apple Watch Series 5 - 40mm (6.1.1) [F76D77EF-0
932-4164-94BB-9FC757420911] (Simulator)
        iPhone 11 Pro Max (13.3) [50C15135-1532-44C5-B82C-B327F88F2712] (Simul
ator)
        iPhone 11 Pro Max (13.3) + Apple Watch Series 5 - 44mm (6.1.1) [D86F0B
D5-4D38-4537-9C8C-2F5C74E404CA] (Simulator)
        iPhone 8 (13.3) [54589698-0C9F-407D-B21A-83432CABB681] (Simulator)
        iPhone 8 Plus (13.3) [509B7103-97DB-4AB9-B829-001190ED4B7E] (Simulator
)
        ...
return deviceStrList
```

调用举例：

```
def get_phone_name_iOS(self):
    deviceName = ""
    deviceStrList = self.get_iOS_deviceStrList()
    # Mo iPhone (11.4.1) [fc9e1f9f2e1339328b9580f826a30fded3bc69ff]
    curDevP = "(?P<deviceName>[\w ]+)\s+\((?P<iOSVersion>[\d\.\.]+)\)\s+\[%s\]" % self.devic
e
    for eachDeviceStr in deviceStrList:
        foundCurDev = re.search(curDevP, eachDeviceStr)
        if foundCurDev:
            deviceName = foundCurDev.group("deviceName")
            iOSVersion = foundCurDev.group("iOSVersion")
            break

    return deviceName
```

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2020-08-09 10:13:38

app

此处整理和app相关的常用代码段。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09 10:13:38

app管理

说明：下面app管理期间用到的iOS的内置的app，其bundle id可参考：[iOS内置app的bundle id](#)

get_PackageActivity_iOS 当前正在运行的app的信息

```
def get_PackageActivity_iOS(self):
    """
    {
        "running" : true,
        "state" : 4,
        "generation" : 0,
        "processArguments" : {
            "env" : {},
            "args" : []
        },
        "title" : "",
        "bundleId" : "com.tencent.xin",
        "label" : "微信",
        "path" : "",
        "name" : "",
        "pid" : 1357
    }
    """
    curAppInfo = self.wdaClient.app_current()
    logging.debug("curAppInfo=%s", curAppInfo)
    return curAppInfo
```

调用：

```
# 获取获取当前活跃app及activity方法
curAppInfo = self.get_PackageActivity_iOS()
curAppStateInt = curAppInfo["state"]
curStateEnum = ApplicationState(curAppStateInt)
logging.debug("curStateEnum=%s", curStateEnum)
bundleId = curAppInfo["bundleId"]
logging.debug("bundleId=%s", bundleId)
curAppName = curAppInfo["label"]
logging.debug("curAppName=%s", curAppName)

package = bundleId
```

获取app状态

```
def iOSGetAppState(self, appBundleId):
    """get iOS app state
```

```

Args:
    appBundleId (str): iOS app bundle id
Returns:
    bool, enum/dict:
        true, app status enum
        false, error info dict
Raises:
"""
curAppState = self.wdaClient.app_state(appBundleId)
logging.debug("curAppState=%s", curAppState)
"""
{
    "value" : 4,
    "sessionId" : "5BBD460B-F420-461D-A5E3-244A74CDF5CE"
}
"""
# # <GenericDict, len() = 3>
# # curAppStateValue = curAppState[0]
# # curAppStatus = curAppState.status
# # curAppSessionId = curAppState.sessionId
# # logging.debug("curAppStatus=%s, curAppSessionId=%s", curAppStatus, curAppSessionId)

# curAppStateValue = curAppState.value
# logging.debug("curAppStateValue=%s", curAppStateValue)
# curStateEnum = ApplicationState(curAppStateValue)
# logging.debug("curStateEnum=%s", curStateEnum)
# return curStateEnum
isGetOk, respInfo = self.processWdaResponse(curAppState)
if isGetOk:
    respValue = respInfo
    curStateEnum = ApplicationState(respValue)
    logging.debug("curStateEnum=%s", curStateEnum)
    respInfo = curStateEnum
return isGetOk, respInfo

```

启动app

```

def iOSLaunchApp(self, appBundleId):
    """Launch iOS app

Args:
    appBundleId (str): iOS app bundle id
Returns:
    bool, None/str:
        true, None
        false, str: error message
Raises:
"""
launchResp = self.wdaClient.app_launch(appBundleId)

```

```

logging.debug("launchResp=%s", launchResp)
isLaunchOk, respInfo = self.processWdaResponse(launchResp)
return isLaunchOk, respInfo

```

停止app

```

def iOSTerminateApp(self, appBundleId):
    """Terminate iOS app

Args:
    appBundleId (str): iOS app bundle id
Returns:
    bool, bool/str:
        true, bool
            True: terminal Ok
        False: terminal fail
            eg: current not running 设置, if terminal, return False
    false, str: error message
Raises:
"""
# isTerminalOk = False
# respInfo = None
# self.wdaClient.session().app_terminate(appBundleId)
# self.wdaClient().app_terminate(appBundleId)
terminateResp = self.wdaClient.app_terminate(appBundleId)
logging.debug("terminateResp=%s", terminateResp)
# respStatus = resp.status
# respValue = resp.value
# respSessionId = resp.sessionId
# logging.info("respStatus=%s, respValue=%s, respSessionId", respStatus, respValue, respSessionId)
# if respStatus == 0:
#     isTerminalOk = True
#     respInfo = None
# else:
#     errInfo = {
#         "status": respStatus,
#         "value": respValue,
#     }
#     respInfo = errInfo
isTerminalOk, respInfo = self.processWdaResponse(terminateResp)
return isTerminalOk, respInfo

```

调用：

```

if isTerminateFirst:
    if isDebugState:
        isGetOk, curState = self.iOSGetAppState(iOS_AppId_Settings)
        logging.info("before terminal: curState=%s", curState)
    # stop before start to avoid current page is not homepage of 设置

```

```
isTerminalOk, respInfo = self.iOSTerminateApp(iOS_AppId_Settings)
logging.info("%s: isTerminalOk=%s, respInfo=%s", iOS_AppId_Settings, isTerminalOk, respInfo)
if isDebugState:
    isGetOk, curState = self.iOSGetAppState(iOS_AppId_Settings)
    logging.info("after terminal: curState=%s", curState)
```

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09 10:13:38

首次登录引导页

首次登录app时，引导页，多次左滑进入app主页

```

def swipeLeftGuideToMain(self):
    """
        处理iOS中app，在首次登录后，引导页，需要多次左滑，最后进入主页
    """

    """
        途虎养车 左滑引导页 1页/3页：
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
        " x="0" y="0" width="414" height="736">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
            x="0" y="0" width="414" height="736">
                <XCUIElementTypeScrollView type="XCUIElementTypeScrollView" enabled="true"
                visible="true" x="0" y="0" width="414" height="736">
                    <XCUIElementTypeImage type="XCUIElementTypeImage" name="guide0" enabled="true"
                    visible="true" x="0" y="0" width="414" height="736"/>
                </XCUIElementTypeScrollView>
                <XCUIElementTypeButton type="XCUIElementTypeButton" value="1" name="loading
                sign nonsel" label="loading sign nonsel" enabled="true" visible="true" x="177" y="630"
                width="10" height="10"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="loading sign
                nonsel" label="loading sign nonsel" enabled="true" visible="true" x="202" y="630" width="10"
                height="10"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="loading sign
                nonsel" label="loading sign nonsel" enabled="true" visible="true" x="227" y="630" width="10"
                height="10"/>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    """

        途虎养车 左滑引导页 2页/3页：
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
        " x="0" y="0" width="414" height="736">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
            x="0" y="0" width="414" height="736">
                <XCUIElementTypeScrollView type="XCUIElementTypeScrollView" enabled="true"
                visible="true" x="0" y="0" width="414" height="736">
                    <XCUIElementTypeImage type="XCUIElementTypeImage" name="guide1" enabled="true"
                    visible="true" x="0" y="0" width="414" height="736"/>
                </XCUIElementTypeScrollView>
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="loading sign
                nonsel" label="loading sign nonsel" enabled="true" visible="true" x="177" y="630" width="10"
                height="10"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton" value="1" name="loading
                sign nonsel" label="loading sign nonsel" enabled="true" visible="true" x="202" y="630" width="10"
                height="10"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="loading sign
                nonsel" label="loading sign nonsel" enabled="true" visible="true" x="227" y="630" width="10"
                height="10"/>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    """

```

```

        </XCUIElementTypeOther>
    </XCUIElementTypeOther>

    途虎养车 左滑引导页 3页/3页 立即进入:
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
    " x="0" y="0" width="414" height="736">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
        " x="0" y="0" width="414" height="736">
            <XCUIElementTypeScrollView type="XCUIElementTypeScrollView" enabled="true"
            visible="true" x="0" y="0" width="414" height="736">
                <XCUIElementTypeImage type="XCUIElementTypeImage" name="guide2" enabled="true"
                visible="true" x="0" y="0" width="414" height="736"/>
            </XCUIElementTypeScrollView>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="loading btn"
            label="loading btn" enabled="true" visible="true" x="141" y="628" width="132" height="30"
            />
        </XCUIElementTypeOther>
    </XCUIElementTypeOther>
    """
    # GuideText = "guide"
    # parentScrollViewClassChain = "/XCUIElementTypeScrollView[`rect.x = 0 AND rect.y = 0
    AND rect.width = %d AND rect.height = %d`]" % (ScreenX, ScreenY)
    # guideImageQuery = {"type": "XCUIElementTypeImage", "nameContains": GuideText, "enabled":
    "true", "x": "0", "y": "0", "width": "%s" % ScreenX, "height": "%s" % ScreenY}
    # guideImageQuery["parent_class_chains"] = [ parentScrollViewClassChain ]
    # isFound, guideImageElement = findElement(curSession, guideImageQuery, timeout=0.1)
    # if isFound:
    #     # foundAndClicked = clickElement(curSession, guideImageElement)
    #     swipeLeft(curSession)

    foundAndSwipeGuideToMain = False

    swipeNum = 0

    guideP = re.compile("guide") # guide0, guide1, guide2
    guideImageChainList = [
        {
            "tag": "XCUIElementTypeOther",
            "attrs": self.FullScreenAttrDict,
        },
        {
            "tag": "XCUIElementTypeScrollView",
            "attrs": self.FullScreenAttrDict,
        },
        {
            "tag": "XCUIElementTypeImage",
            "attrs": {"name": guideP, **self.FullScreenAttrDict}
        },
    ],
]

while True:
    curPageXml = self.get_page_source()
    soup = CommonUtils.xmlToSoup(curPageXml)

```

```

guideImageSoup = CommonUtils.bsChainFind(soup, guideImageChainList)
if not guideImageSoup:
    break

parentScrollViewSoup = guideImageSoup.parent
if not parentScrollViewSoup:
    break

validButtonSoupList = []

nextSiblingList = parentScrollViewSoup.next_siblings
for eachNextSiblingSoup in nextSiblingList:
    if hasattr(eachNextSiblingSoup, "attrs"):
        soupAttrDict = eachNextSiblingSoup.attrs # {'enabled': 'true', 'height': '30', 'label': 'loading btn', 'name': 'loading btn', 'type': 'XCUIElementTypeButton', 'visible': 'true', 'width': '132', 'x': '141', 'y': '628'}
        soupType = soupAttrDict.get("type") # 'XCUIElementTypeButton'
        soupName = soupAttrDict.get("name") # 'loading btn'
        soupVisible = soupAttrDict.get("visible") # 'true'
        isButton = soupType == "XCUIElementTypeButton"
        # isLoadingName = bool(re.search(soupName, "loading"))
        isLoadingName = bool(re.search("loading", soupName))
        isVisible = soupVisible == "true"
        isValid = isButton and isLoadingName and isVisible
        if isValid:
            validButtonSoupList.append(eachNextSiblingSoup)

if validButtonSoupList:
    validButtonNum = len(validButtonSoupList)
    if validButtonNum == 1:
        # end page, click button, into main page
        lastPageButtonSoup = validButtonSoupList[0]
        self.clickElementCenterPosition(lastPageButtonSoup)
        foundAndSwipeGuideToMain = True
        break
    elif validButtonNum > 1:
        swipeNum += 1
        # not end, should continue to swipe left
        self.swipe("SwipeLeft")
        logging.info("Swipe left for guide page %d", swipeNum)

return foundAndSwipeGuideToMain

```

对应页面：

- 恒易贷

- 第一页：

-

- 最后一页：

-

- 途虎养车
 - 第一页
 -
 - 第二页
 -
 - 第三页
 -

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09 10:13:38

微信相关

判断是否处于微信

```

def iOSIsInWeixin(self):
    tabBarClassChain = "/XCUIElementTypeTabBar[`rect.width = %d`]" % self.X
    """
        <XCUIElementTypeTabBar type="XCUIElementTypeTabBar" enabled="true" visible="true"
    x="0" y="680" width="414" height="56">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
    " x="0" y="680" width="414" height="56">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
true" x="0" y="680" width="414" height="56">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visib
le="false" x="0" y="680" width="414" height="56"/>
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visib
le="false" x="0" y="680" width="414" height="56"/>
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visib
le="true" x="0" y="680" width="414" height="56"/>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
            <XCUIElementTypeButton type="XCUIElementTypeButton" value="2" name="微信" la
bel="微信" enabled="true" visible="true" x="2" y="681" width="100" height="55"/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="通讯录" label="通讯录"
enabled="true" visible="true" x="106" y="681" width="99" height="55"/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" value="未读" name="发现" la
bel="发现" enabled="true" visible="true" x="209" y="681" width="100" height="55"/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="我" label="我" enab
le="true" visible="true" x="313" y="681" width="99" height="55"/>
        </XCUIElementTypeTabBar>
    """
    weixinTabQuery = {"type": "XCUIElementTypeButton", "name": "微信", "label": "微信", "en
abled": "true"}
    weixinTabQuery["parent_class_chains"] = [tabBarClassChain]
    isFoundWeixin, respInfo = self.findElement(query weixinTabQuery, timeout 0.2)
    iOSIsInWeixin = isFoundWeixin
    return iOSIsInWeixin

```

调用：

```
beInWeixin = self.iOSIsInWeixin()
```

get_navigationBar_bounds 计算微信顶部系统导航栏的区域
bounds

```

# def get_navigationBar_bounds(self, pageSrcXml):
def get_navigationBar_bounds(self):
    """
        计算微信顶部系统导航栏的区域bounds
    """
    bounds = None
    """
        微信 顶部 导航栏:
        <XCUIElementTypeNavigationBar type="XCUIElementTypeNavigationBar" name="公众号"
enabled="true" visible="true" x="0" y="20" width="375" height="44">
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="通讯录" label="通
讯录" enabled="true" visible="true" x="10" y="20" width="64" height="44"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="公众号"
name="公众号" label="公众号" enabled="true" visible="true" x="187" y="24" width="1" height="36"/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="添加" label="添加"
" enabled="true" visible="true" x="317" y="20" width="58" height="44"/>
        </XCUIElementTypeNavigationBar>
    """
    naviBarQuery = {"type": "XCUIElementTypeNavigationBar", "enabled": "true", "x": "0"}
    isFound, naviBarElement = self.findElement(naviBarQuery)
    if isFound:
        isVisible = naviBarElement.visible
        if isVisible:
            curRect = naviBarElement.bounds
            logging.debug("curRect=%s", curRect)
            x0 = curRect.x
            y0 = curRect.y
            x1 = curRect.x1
            y1 = curRect.y1
            bounds = [x0, y0, x1, y1] # [0, 20, 375, 64]

    logging.debug("bounds=%s", bounds)
    return bounds

```

调用举例：

```

def calcHeadY(curSession):
    bounds = get_navigationBar_bounds(curSession)
    y1 = bounds[-1]
    headY = y1
    return headY

```

安全模式

背景

在 facebook-wda 调试期间，通过代码控制微信app的退出和启动，多次如此调试后就会导致微信检测到以为发生了多次的异常，所以会进入安全模式。

此时只能根据提示，一次次点击 下一步 ，才能最终退出 安全模式：









详见：

【已解决】自动抓包工具适配iOS：当前检测出微信连续异常，你可以尝试一下方法修复 下一步

代码： iOSWeixinExceptionNextStep

自动检测是否处于 安全模式，并自动点击直到退出 安全模式 的代码如下，供参考：

```

def iOSMakessureIntoWeixin(self):
    """Makessure into weixin main page
    if exception, process it, until into weixin page
    """
    # maxRetryNum = 3
    maxRetryNum = 5
    beInWeixin = self.iOSisInWeixin()

    while (not beInWeixin) and (maxRetryNum > 0):
        beInWeixin = self.iOSisInWeixin()
        if not beInWeixin:
            # try process for exception
            foundAndProcessedException = self.iOSWeixinExceptionNextStep()
            if foundAndProcessedException:
                beInWeixin = self.iOSisInWeixin()

    maxRetryNum -= 1

    return beInWeixin

def iOSWeixinExceptionNextStep(self):
    foundAndClicked = False

    scrollViewClassChain = "/XCUIElementTypeScrollView[`rect.width = %d AND rect.height = %d`]" % (self.X, self.totalY)

    ButtonLabelNextStep = "下一步"
    ButtonLabelIntoWeixin = "进入微信"

```

```

# nextStepButtonQuery = {"type": "XCUIElementTypeButton", "label": "下一步", "enabled": "true"}
# nextStepButtonQuery["parent_class_chains"] = [scrollViewClassChain]

# intoWeixinButtonQuery = {"type": "XCUIElementTypeButton", "label": "进入微信", "enabled": "true"}
# intoWeixinButtonQuery["parent_class_chains"] = [scrollViewClassChain]

"""
<XCUIElementTypeScrollView type="XCUIElementTypeScrollView" enabled="true" visible="true" x="0" y="0" width="414" height="736">
    <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="false" x="147" y="106" width="120" height="120"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="当前检测出微信连续异常，你可以尝试以下方法修复：" name="当前检测出微信连续异常，你可以尝试以下方法修复：" label="当前检测出微信连续异常，你可以尝试以下方法修复：" enabled="true" visible="true" x="18" y="262" width="378" height="39"/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="下一步" label="下一步" enabled="true" visible="true" x="18" y="319" width="378" height="47">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="下一步" name="下一步" label="下一步" enabled="true" visible="true" x="179" y="331" width="56" height="23"/>
            </XCUIElementTypeButton>
        </XCUIElementTypeScrollView>
"""

# continuousExceptionQuery = {"type": "XCUIElementTypeStaticText", "value": "当前检测出微信连续异常，你可以尝试以下方法修复：" , "enabled": "true"}
# continuousExceptionQuery["parent_class_chains"] = [scrollViewClassChain]
# isFoundContinuousException, respInfo = self.findElement(query=continuousExceptionQuery, timeout=0.5)
# if isFoundContinuousException:
#     isFoundNextStep, respInfo = self.findElement(query=nextStepButtonQuery)
#     if isFoundNextStep:
#         nextStepElement = respInfo
#         foundAndClicked = self.clickElement(nextStepElement)

"""
<XCUIElementTypeScrollView type="XCUIElementTypeScrollView" enabled="true" visible="true" x="0" y="0" width="414" height="736">
    <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="false" x="147" y="106" width="120" height="120"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="建议你重启手机，避免微信启动异常" name="建议你重启手机，避免微信启动异常" label="建议你重启手机，避免微信启动异常" enabled="true" visible="true" x="18" y="262" width="378" height="15"/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="重启手机" label="重启手机" enabled="true" visible="true" x="18" y="285" width="378" height="47">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="重启手机" name="重启手机" label="重启手机" enabled="true" visible="true" x="170" y="297" width="74" height="23"/>
            </XCUIElementTypeButton>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="不重启手机，进入下一步" name="不重启手机，进入下一步" label="不重启手机，进入下一步" enabled="true" visible="true">

```

```

        ="true" x="18" y="356" width="378" height="15"/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="下一步" label="下一步"
enabled="true" visible="true" x="18" y="379" width="378" height="47">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="下一步"
name="下一步" label="下一步" enabled="true" visible="true" x="179" y="391" width="56" height=
"23"/>
            </XCUIElementTypeButton>
            <XCUIElementTypeOther type="XCUIElementTypeOther" value="0%" name="垂直滚动条,
1页" label="垂直滚动条, 1页" enabled="true" visible="true" x="381" y="64" width="30" height=
"672"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" value="0%" name="水平滚动条,
1页" label="水平滚动条, 1页" enabled="true" visible="true" x="0" y="703" width="414" height=
"30"/>
        </XCUIElementTypeScrollView>
    """
    # # if not foundAndClicked:
    # rebootWeixinQuery = {"type": "XCUIElementTypeStaticText", "value": "建议你重启手机，避
免微信启动异常", "enabled": "true"}
    # rebootWeixinQuery["parent_class_chains"] = [scrollViewClassChain]
    # isFoundRebootWeixin, respInfo = self.findElement(query=rebootWeixinQuery, timeout=0.
5)
    # if isFoundRebootWeixin:
    #     isFoundNextStep, respInfo = self.findElement(query=nextStepButtonQuery)
    #     if isFoundNextStep:
    #         nextStepElement = respInfo
    #         foundAndClicked = self.clickElement(nextStepElement)

    """
        <XCUIElementTypeScrollView type="XCUIElementTypeScrollView" enabled="true" visible
="true" x="0" y="0" width="414" height="736">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
" x="0" y="64" width="414" height="243">
                <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="
false" x="115" y="94" width="184" height="183"/>
                <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="
false" x="115" y="94" width="184" height="183"/>
            </XCUIElementTypeOther>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="清理缓存会清
理你的手机本地缓存文件，但不会清理你的消息数据，使用后需要重新登录微信" name="清理缓存会清
理你的手机本地缓存文件，但不会清理你的消息数据，使用后需要重新登录微信" label="清理缓存会清理你的手
机本地缓存文件，但不会清理你的消息数据，使用后需要重新登录微信" enabled="true" visible="true" x="18" y=
"330" width="378" height="29"/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="清理缓存" label="清理
缓存" enabled="true" visible="true" x="18" y="367" width="378" height="47">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="清理缓存
" name="清理缓存" label="清理缓存" enabled="true" visible="true" x="170" y="379" width="74"
height="23"/>
            </XCUIElementTypeButton>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="不清理缓存,
进入下一步" name="不清理缓存, 进入下一步" label="不清理缓存, 进入下一步" enabled="true" visible
="true" x="18" y="438" width="378" height="15"/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="下一步" label="下一步"
enabled="true" visible="true" x="18" y="461" width="378" height="47">

```

```

                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="下一步"
name="下一步" label="下一步" enabled="true" visible="true" x="179" y="473" width="56" height
="23"/>
            </XCUIElementTypeButton>
            <XCUIElementTypeOther type="XCUIElementTypeOther" value="0%" name="垂直滚动条,
1页" label="垂直滚动条, 1页" enabled="true" visible="true" x="381" y="64" width="30" height=
"672"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" value="0%" name="水平滚动条,
1页" label="水平滚动条, 1页" enabled="true" visible="true" x="0" y="703" width="414" height=
"30"/>
        </XCUIElementTypeScrollView>
        """
# # if not foundAndClicked:
# clearCacheQuery = {"type": "XCUIElementTypeStaticText", "value": "清理缓存会清理你的手
机本地缓存文件, 但不会清理你的消息数据, 使用后需要重新登录微信", "enabled": "true"}
# clearCacheQuery["parent_class_chains"] = [scrollViewClassChain]
# isFoundClearCache, respInfo = self.findElement(query=clearCacheQuery, timeout=0.5)
# if isFoundClearCache:
#     isFoundNextStep, respInfo = self.findElement(query=nextStepButtonQuery)
#     if isFoundNextStep:
#         nextStepElement = respInfo
#         foundAndClicked = self.clickElement(nextStepElement)

"""
<XCUIElementTypeScrollView type="XCUIElementTypeScrollView" enabled="true" visible
="true" x="0" y="0" width="414" height="736">
    <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="fals
e" x="147" y="106" width="120" height="120"/>
    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="如果问题还没
解决, 你可以上传手机日志文件, 协助技术人员解决问题。所上传的文件不会包含聊天记录等私人内容, 且不会被
对外传播" name="如果问题还没解决, 你可以上传手机日志文件, 协助技术人员解决问题。所上传的文件不会包
含聊天记录等私人内容, 且不会被对外传播" label="如果问题还没解决, 你可以上传手机日志文件, 协助技术人
员解决问题。所上传的文件不会包含聊天记录等私人内容, 且不会被对外传播" enabled="true" visible="tru
e" x="18" y="262" width="378" height="29"/>
    <XCUIElementTypeButton type="XCUIElementTypeButton" name="上传文件" label="上传
文件" enabled="true" visible="true" x="18" y="298" width="378" height="47">
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="上传文件
" name="上传文件" label="上传文件" enabled="true" visible="true" x="170" y="310" width="74"
height="23"/>
    </XCUIElementTypeButton>
    <XCUIElementTypeButton type="XCUIElementTypeButton" name="显示上传文件列表" labe
l="显示上传文件列表" enabled="true" visible="true" x="17" y="353" width="100" height="15">
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="显示上传
文件列表" name="显示上传文件列表" label="显示上传文件列表" enabled="true" visible="true" x="17"
y="352" width="100" height="16"/>
    </XCUIElementTypeButton>
    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="不上传文件,
进入下一步" name="不上传文件, 进入下一步" label="不上传文件, 进入下一步" enabled="true" visible
="true" x="18" y="391" width="378" height="15"/>
    <XCUIElementTypeButton type="XCUIElementTypeButton" name="下一步" label="下一
步" enabled="true" visible="true" x="18" y="413" width="378" height="47">
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="下一步"
name="下一步" label="下一步" enabled="true" visible="true" x="179" y="425" width="56" height
="23"/>
    </XCUIElementTypeButton>

```

```

        ="23"/>
    </XCUIElementTypeButton>
    <XCUIElementTypeOther type="XCUIElementTypeOther" value="0%" name="垂直滚动条,
1页" label="垂直滚动条, 1页" enabled="true" visible="true" x="381" y="64" width="30" height=
"672"/>
    <XCUIElementTypeOther type="XCUIElementTypeOther" value="0%" name="水平滚动条,
1页" label="水平滚动条, 1页" enabled="true" visible="true" x="0" y="703" width="414" height=
"30"/>
    </XCUIElementTypeScrollView>
"""

"""

<XCUIElementTypeScrollView type="XCUIElementTypeScrollView" enabled="true" visible
="true" x="0" y="0" width="414" height="736">
    <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="false"
x="147" y="106" width="120" height="120"/>
    <XCUIElementTypeButton type="XCUIElementTypeButton" enabled="true" visible="true"
x="147" y="106" width="120" height="120"/>
    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="如果使用过
程中还出现问题, 建议你重启手机, 更新系统, 或者联系我们的客服人员。 当前设备标
识: 7a3e060db5c52291***"
***" name="如果使用过程中还出现问题, 建议你重启手机, 更新系统, 或者联系我们的客服人员。 当前设备标
识: 7a3e060db5c52291*****" label="如果使用过程中还出现问题, 建议你重启手机, 更新系统, 或者联系
我们的客服人员。 当前设备标识: 7a3e060db5c52291*****" enabled="true" visible="true" x="18" y
="262" width="378" height="58"/>
    <XCUIElementTypeButton type="XCUIElementTypeButton" name="进入微信" label="进入
微信" enabled="true" visible="true" x="18" y="338" width="378" height="47">
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="进入微信
" name="进入微信" label="进入微信" enabled="true" visible="true" x="170" y="350" width="74"
height="23"/>
    </XCUIElementTypeButton>
    <XCUIElementTypeOther type="XCUIElementTypeOther" value="0%" name="垂直滚动条,
1页" label="垂直滚动条, 1页" enabled="true" visible="true" x="381" y="64" width="30" height=
"672"/>
    <XCUIElementTypeOther type="XCUIElementTypeOther" value="0%" name="水平滚动条,
1页" label="水平滚动条, 1页" enabled="true" visible="true" x="0" y="703" width="414" height=
"30"/>
    </XCUIElementTypeScrollView>
"""

EachStepNoticeList = [
    # ("当前检测出微信连续异常, 你可以尝试以下方法修复: ", ButtonLabelNextStep),
    ("当前检测出微信连续异常", ButtonLabelNextStep),
    ("建议你重启手机, 避免微信启动异常", ButtonLabelNextStep),
    ("清理缓存会清理你的手机本地缓存文件, 但不会清理你的消息数据, 使用后需要重新登录微信", But
tonLabelNextStep),
    ("如果问题还没解决, 你可以上传手机日志文件, 协助技术人员解决问题。所上传的文件不会包含聊天
记录等私人内容, 且不会被对外传播", ButtonLabelNextStep),
    ("如果使用过程中还出现问题, 建议你重启手机, 更新系统, 或者联系我们的客服人员。", ButtonLa
belIntoWeixin),
]
for (curStepNotice, buttonLabel) in EachStepNoticeList:
    # curNoticeQuery = {"type": "XCUIElementTypeStaticText", "value": curStepNotice, "e
nabled": "true"}  


```

```
# curNoticeQuery = {"type": "XCUIElementTypeStaticText", "value_part": curStepNotice, "enabled": "true"}
    curNoticeQuery = {"type": "XCUIElementTypeStaticText", "valueContains": curStepNotice, "enabled": "true"}
    curNoticeQuery["parent_class_chains"] = [scrollViewClassChain]
    isFoundCurNotice, respInfo = self.findElement(query=curNoticeQuery, timeout=0.5)
    if isFoundCurNotice:
        # isFoundNextStep, respInfo = self.findElement(query=nextStepButtonQuery)
        # isFoundButton, respInfo = self.findElement(query=buttonQuery)
        buttonQuery = {"type": "XCUIElementTypeButton", "label": buttonLabel, "enabled": "true"}
        buttonQuery["parent_class_chains"] = [scrollViewClassChain]
        isFoundButton, respInfo = self.findElement(query=buttonQuery)
        if isFoundButton:
            buttonElement = respInfo
            clickOk = self.clickElement(buttonElement)
            if clickOk:
                foundAndClicked = clickOk

    return foundAndClicked
```



crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09 10:13:38

微信公众号

此处整理微信中关于微信公众号的部分的常见代码段。

isPublicAccountSearchPage_iOS 判断是否处于微信公众号搜索页

```
def isPublicAccountSearchPage_iOS(self, page):
    """Check whether current page is Weixin Public account search page"""
    isPublicAccountSearch = False
    curInputValue = ""

    """
    is search:
        <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true"
    " x="0" y="0" width="414" height="70">
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="返回" label="返回"
    " enabled="true" visible="true" x="0" y="20" width="33" height="50"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
    true" x="33" y="20" width="381" height="56">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visib
    le="true" x="33" y="20" width="381" height="50">
                    <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" v
    isible="false" x="33" y="0" width="381" height="70"/>
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" v
    isible="true" x="33" y="20" width="381" height="50">
                        <XCUIElementTypeSearchField type="XCUIElementTypeSearchField"
    name="搜索" label="搜索" enabled="true" visible="true" x="41" y="27" width="319"
    height="36"/>
                    <XCUIElementTypeButton type="XCUIElementTypeButton" name="取消"
    label="取消" enabled="true" visible="true" x="371" y="29" width="35" height="30">
                        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText"
    value="取消" name="取消" label="取消" enabled="true" visible="true" x="371" y="33"
    width="35" height="22"/>
                    </XCUIElementTypeButton>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    </XCUIElementTypeImage>

        <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true"
    " x="0" y="0" width="414" height="70">
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="返回" label="返回"
    " enabled="true" visible="true" x="0" y="20" width="33" height="50"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
    true" x="33" y="20" width="381" height="56">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visib
    le="true" x="33" y="20" width="381" height="50">
                    <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" v
    isible="
```

```

        false" x="33" y="0" width="381" height="70"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="33" y="20" width="381" height="50">
                <XCUIElementTypeSearchField type="XCUIElementTypeSearchField" value="g_h_cfcfcee032cc" name="搜索" label="搜索" enabled="true" visible="true" x="41" y="27" width="319" height="36">
                    <XCUIElementTypeButton type="XCUIElementTypeButton" name="清除文本" label="清除文本" enabled="true" visible="true" x="335" y="35" width="20" height="20"/>
                </XCUIElementTypeSearchField>
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="取消" label="取消" enabled="true" visible="true" x="371" y="29" width="35" height="30">
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="取消" name="取消" label="取消" enabled="true" visible="true" x="371" y="33" width="35" height="22"/>
                </XCUIElementTypeButton>
            </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
        </XCUIElementTypeImage>

        <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="0" y="0" width="414" height="70">
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="返回" label="返回" enabled="true" visible="true" x="0" y="20" width="33" height="50"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="33" y="20" width="381" height="56">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="33" y="20" width="381" height="50">
                    <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="false" x="33" y="0" width="381" height="70"/>
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="33" y="20" width="381" height="50">
                        <XCUIElementTypeSearchField type="XCUIElementTypeSearchField" value="g_h_cfcfcee032cc" name="搜索" label="搜索" enabled="true" visible="true" x="41" y="27" width="319" height="36">
                            <XCUIElementTypeButton type="XCUIElementTypeButton" name="清除文本" label="清除文本" enabled="true" visible="false" x="335" y="35" width="20" height="20"/>
                        </XCUIElementTypeSearchField>
                        <XCUIElementTypeButton type="XCUIElementTypeButton" name="取消" label="取消" enabled="true" visible="true" x="371" y="29" width="35" height="30">
                            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="取消" name="取消" label="取消" enabled="true" visible="true" x="371" y="33" width="35" height="22"/>
                        </XCUIElementTypeButton>
                    </XCUIElementTypeOther>
                    </XCUIElementTypeOther>
                </XCUIElementTypeOther>
                </XCUIElementTypeImage>
            </XCUIElementTypeOther>
        </XCUIElementTypeImage>
    """
    soup = CommonUtils.xmlToSoup(page)
    widthStr = str(self.X)
    foundImage = soup.find(
        'XCUIElementTypeImage',

```

```

        attrs {"type": "XCUIElementTypeImage", "enabled": "true", "x": "0", "y": "0", "width": widthStr},
    )
    if foundImage:
        foundSearchField = foundImage.find("XCUIElementTypeSearchField",
            attrs={"type": "XCUIElementTypeSearchField", "name": "搜索", "label": "搜索", "enabled": "true"})
    )
    if foundSearchField:
        isPublicAccountSearch = True
        # curInputValue = foundSearchField.attrs["value"]
        curInputValue = foundSearchField.attrs.get("value")

    return isPublicAccountSearch, curInputValue

```

isPublicAccountFocusOrIntoPage_iOS

```

def isPublicAccountFocusOrIntoPage_iOS(self, page):
    """Check whether current page is Weixin Public focus or enter into page"""
    isPublicAccountFocusOrInto = False

    """
        is Focus:
            <XCUIElementTypeNavigationBar type="XCUIElementTypeNavigationBar" name="通讯录"
            enabled="true" visible="true" x="0" y="20" width="414" height="44">
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="返回" label="返回"
            " enabled="true" visible="true" x="20" y="20" width="30" height="44"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
            false" x="206" y="24" width="2" height="36"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="搜索" label="搜索"
            " enabled="true" visible="true" x="306" y="20" width="40" height="44"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="更多" label="更多"
            " enabled="true" visible="true" x="354" y="20" width="40" height="44"/>
            </XCUIElementTypeNavigationBar>

            <XCUIElementTypeTable type="XCUIElementTypeTable" enabled="true" visib
            le="true" x="0" y="0" width="414" height="736">
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="头像" lab
            el="头像" enabled="true" visible="true" x="20" y="76" width="69" height="69"/>
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value=
            "动卡空间" name="动卡空间" label="动卡空间" enabled="true" visible="false" x="108" y="74" wid
            th="82" height="24"/>
                <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" v
            isible="false" x="194" y="78" width="16" height="17"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="分享用卡
            活动, 传播正能量" label="分享用卡活动, 传播正能量" enabled="true" visible="true" x="108" y="10
            8" width="270" height="17">
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" va
            lue="分享用卡活动, 传播正能量" name="分享用卡活动, 传播正能量" label="分享用卡活动, 传播正能量" e
            nabled="true" visible="false" x="108" y="108" width="172" height="17"/>
            </XCUIElementTypeTable>
        </XCUIElementTypeScrollView>
    </XCUIElementTypePage>

```

```

        </XCUIElementTypeButton>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="2位朋友关注" name="2位朋友关注" label="2位朋友关注" enabled="true" visible="true" x="108" y="135" width="82" height="17"/>
        <XCUIElementTypeButton type="XCUIElementTypeButton" name="关注公众号" label="关注公众号" enabled="true" visible="true" x="166" y="177" width="82" height="20">
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="关注公众号" name="关注公众号" label="关注公众号" enabled="true" visible="false" x="166" y="176" width="83" height="21"/>
        </XCUIElementTypeButton>
        <XCUIElementTypeButton type="XCUIElementTypeButton" name="进入公众号" label="进入公众号" enabled="true" visible="false" x="0" y="177" width="82" height="20">
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="进入公众号" name="进入公众号" label="进入公众号" enabled="true" visible="false" x="0" y="176" width="82" height="21"/>
        </XCUIElementTypeButton>
        <XCUIElementTypeButton type="XCUIElementTypeButton" name="不再关注" label="不再关注" enabled="true" visible="false" x="0" y="177" width="66" height="20">
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="不再关注" name="不再关注" label="不再关注" enabled="true" visible="false" x="-1" y="176" width="67" height="21"/>
        </XCUIElementTypeButton>

    is enter into:
    <XCUIElementTypeNavigationBar type="XCUIElementTypeNavigationBar" name="通讯录" enabled="true" visible="true" x="0" y="20" width="414" height="44">
        <XCUIElementTypeButton type="XCUIElementTypeButton" name="返回" label="返回" enabled="true" visible="true" x="20" y="20" width="30" height="44"/>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="206" y="24" width="2" height="36"/>
        <XCUIElementTypeButton type="XCUIElementTypeButton" name="搜索" label="搜索" enabled="true" visible="true" x="306" y="20" width="40" height="44"/>
        <XCUIElementTypeButton type="XCUIElementTypeButton" name="更多" label="更多" enabled="true" visible="true" x="354" y="20" width="40" height="44"/>
    </XCUIElementTypeNavigationBar>

    <XCUIElementTypeTable type="XCUIElementTypeTable" enabled="true" visible="true" x="0" y="0" width="414" height="736">
        <XCUIElementTypeButton type="XCUIElementTypeButton" name="头像" label="头像" enabled="true" visible="true" x="20" y="76" width="69" height="69"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="动卡空间" name="动卡空间" label="动卡空间" enabled="true" visible="false" x="108" y="74" width="82" height="24"/>
        <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="false" x="194" y="78" width="16" height="17"/>
        <XCUIElementTypeButton type="XCUIElementTypeButton" name="分享用卡活动,传播正能量" label="分享用卡活动,传播正能量" enabled="true" visible="true" x="108" width="270" height="17">
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="分享用卡活动,传播正能量" name="分享用卡活动,传播正能量" label="分享用卡活动,传播正能量" enabled="true" visible="false" x="108" y="108" width="172" height="17"/>
        </XCUIElementTypeButton>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="2位"

```

```

朋友关注" name="2位朋友关注" label="2位朋友关注" enabled="true" visible="true" x="108" y="135
" width="82" height="17"/>
    <XCUIElementTypeButton type="XCUIElementTypeButton" enabled="true" vis
ible="false" x="0" y="177" width="0" height="0">
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" enable
d="true" visible="false" x="0" y="177" width="0" height="0"/>
    </XCUIElementTypeButton>
    <XCUIElementTypeButton type="XCUIElementTypeButton" name="进入公众号" l
abel="进入公众号" enabled="true" visible="true" x="95" y="177" width="82" height="20">
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value=
"进入公众号" name="进入公众号" label="进入公众号" enabled="true" visible="false" x="95" y="17
6" width="83" height="21"/>
    </XCUIElementTypeButton>
    <XCUIElementTypeButton type="XCUIElementTypeButton" name="不再关注" lab
el="不再关注" enabled="true" visible="true" x="237" y="177" width="66" height="20">
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value=
"不再关注" name="不再关注" label="不再关注" enabled="true" visible="false" x="236" y="176" wi
dth="67" height="21"/>
    </XCUIElementTypeButton>

"""
soup = CommonUtils.xmlToSoup(page)
foundNaviBar = soup.find(
    'XCUIElementTypeNavigationBar',
    attrs {"type": "XCUIElementTypeNavigationBar", "enabled": "true"},
)
# logging.debug("foundNaviBar=%s", foundNaviBar)
if foundNaviBar:
    foundTypeOther = foundNaviBar.find("XCUIElementTypeOther",
        attrs {"type": "XCUIElementTypeOther", "enabled": "true"})
    )
    logging.debug("foundTypeOther=%s", foundTypeOther)
    if foundTypeOther:
        # typeOtherName = foundTypeOther.attrs["name"]
        typeOtherName = foundTypeOther.attrs.get("name")
        logging.debug("typeOtherName=%s", typeOtherName)
        isTypeOtherNameNotEmpty = bool(typeOtherName)
        isTypeOtherNameEmpty = not isTypeOtherNameNotEmpty
        logging.debug("isTypeOtherNameEmpty=%s", isTypeOtherNameEmpty)
        if isTypeOtherNameEmpty:
            foundIntoAccount = soup.find(
                'XCUIElementTypeButton',
                attrs {"type": "XCUIElementTypeButton", "enabled": "true", "name": "进入公
众号", "label": "进入公众号"}, )
            )
            logging.debug("foundIntoAccount=%s", foundIntoAccount)
            if foundIntoAccount:
                prevSiblingList = foundIntoAccount.previous_siblings
                logging.debug("prevSiblingList=%s", prevSiblingList)
                curAccountZhcnName = self.account_zw
                TypeStaticText = "XCUIElementTypeStaticText"
                for eachPrevSibling in prevSiblingList:
                    # curType = eachPrevSibling.attrs["type"]

```

```

        # curName = eachPrevSibling.attrs["name"]
        # curType = eachPrevSibling.attrs.get("type")
        # curName = eachPrevSibling.attrs.get("name")
        if hasattr(eachPrevSibling, "attrs"):
            curType = eachPrevSibling.attrs.get("type")
            curName = eachPrevSibling.attrs.get("name")
            if (curType == TypeStaticText) and (curName == curAccountZhcnName):
                isPublicAccountFocusOrInto = True
            else:
                logging.debug("eachPrevSibling=%s no attrs", eachPrevSibling)

        logging.debug("isPublicAccountFocusOrInto=%s", isPublicAccountFocusOrInto)
    return isPublicAccountFocusOrInto

```

调用：

```

def isCurPageInOffline_iOS(self, page):
    """Check whether current page is belong to (unexpeced) offline page

    offline page include:
    * 微信首页
    * 微信通讯录
    * 微信公众号列表页
    * 微信公众号搜索-待输入
    * 微信公众号搜索-输入公众号ID
    * 微信公众号搜索-搜索结果
    * 微信公众号-关注公众号
    * 微信公众号-进入公众号
    """
    isOffline = False

    OfflinePageNavBarNameList = [
        "微信",
        "通讯录",
        "公众号",
    ]
    hasNavBar, naviBarName = self.isPageHasNavBar_iOS(page)
    if hasNavBar:
        if naviBarName in OfflinePageNavBarNameList:
            # is in some weixin page
            isOffline = True

    if not isOffline:
        isPublicAccountSearch, curInputValue = self.isPublicAccountSearchPage_iOS(page)
        if isPublicAccountSearch:
            # is in public account search page
            isOffline = True

    if not isOffline:
        isPublicAccountFocusOrIntoPage = self.isPublicAccountFocusOrIntoPage_iOS(page)
        if isPublicAccountFocusOrIntoPage:

```

```
# is in public account focus or enter into page
isOffline = True

return isOffline
```

findWeixinPublicAccountZhcnFullName 微信公众号搜索结果列表页 查找 微信公众号的中文（全）名

对于页面：





从中查找出 公众号的中文名 (的全称)

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

Args:
    soup (soup): soup of current page xml
Returns:
    public account zh-CN full name
Raises:
    """
# accountZhcnTextSoup = None
accountZhcnFullName = ""
parentNodeLocator = None

"""
搜索结果中文名节点是Text
    <XCUIElementTypeOther type="XCUIElementTypeOther" name="搜一搜" label="搜一搜" e
nabled="true" visible="true" x="0" y="70" width="414" height="666">
        <XCUIElementTypeOther type="XCUIElementTypeOther" value="2" name="公众号" l
```

```

        abel="公众号" enabled="true" visible="true" x="16" y="83" width="398" height="23">
            <XCUIElementTypeOther type="XCUIElementTypeOther" value="2" enabled="t
rue" visible="true" x="16" y="83" width="52" height="23">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value=
"2" name="公众号" label="公众号" enabled="true" visible="true" x="16" y="83" width="52" heig
ht="22"/>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
false" x="16" y="129" width="60" height="60"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
true" x="88" y="126" width="310" height="23">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="动卡
空间" name="动卡空间" label="动卡空间" enabled="true" visible="true" x="88" y="126" width="7
0" height="22"/>
            </XCUIElementTypeOther>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
true" x="88" y="151" width="310" height="21">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="分享
用卡活动, 传播正能量" name="分享用卡活动, 传播正能量" label="分享用卡活动, 传播正能量" enabl
ed="true" visible="true" x="88" y="152" width="184" height="19"/>
            </XCUIElementTypeOther>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
true" x="88" y="178" width="266" height="20">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="微信
号: " name="微信号: " label="微信号: " enabled="true" visible="true" x="88" y="178" width="5
8" height="19"/>
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="gh_
cfcfcee032cc" name="gh_cfcfcee032cc" label="gh_cfcfcee032cc" enabled="true" visible="true"
x="145" y="178" width="112" height="19"/>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    
```

搜索结果中文名节点是Other, 其下是多个Text节点:

```

        <XCUIElementTypeOther type="XCUIElementTypeOther" name="搜一搜" label="搜一搜" e
nabled="true" visible="true" x="0" y="70" width="375" height="600">
            <XCUIElementTypeOther type="XCUIElementTypeOther" value="2" name="公众号" l
abel="公众号" enabled="true" visible="true" x="16" y="83" width="359" height="23">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="2"
name="公众号" label="公众号" enabled="true" visible="true" x="16" y="83" width="52" height="2
3"/>
            </XCUIElementTypeOther>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
false" x="16" y="129" width="60" height="60"/>
                <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="
true" x="343" y="128" width="16" height="16"/>
                <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="
true" x="324" y="128" width="16" height="16"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
true" x="88" y="126" width="271" height="23">
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="牛尔
" name="牛尔" label="牛尔" enabled="true" visible="true" x="88" y="126" width="35" height="2
2"/>
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="Tma

```

```

    11" name="Tmall" label="Tmall" enabled="true" visible="true" x="122" y="126" width="41" height="22"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="旗舰店" name="旗舰店" label="旗舰店" enabled="true" visible="true" x="162" y="126" width="53" height="22"/>
    </XCUIElementTypeOther>
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="88" y="151" width="271" height="21">
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="牛尔亲研天猫官方旗舰店" name="牛尔亲研天猫官方旗舰店" label="牛尔亲研天猫官方旗舰店" enabled="true" visible="true" x="88" y="152" width="169" height="19"/>
    </XCUIElementTypeOther>
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="88" y="178" width="271" height="19">
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="微信号:" name="微信号:" label="微信号:" enabled="true" visible="true" x="88" y="178" width="58" height="18"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="niuer-tmall" name="niuer-tmall" label="niuer-tmall" enabled="true" visible="true" x="145" y="178" width="70" height="18"/>
    </XCUIElementTypeOther>

```

公众号中文名全部是绿色的：

```

        <XCUIElementTypeOther type="XCUIElementTypeOther" name="搜一搜" label="搜一搜" enabled="true" visible="true" x="0" y="70" width="414" height="666">
            <XCUIElementTypeOther type="XCUIElementTypeOther" value="2" name="公众号" label="公众号" enabled="true" visible="true" x="16" y="83" width="398" height="23">
                <XCUIElementTypeOther type="XCUIElementTypeOther" value="2" enabled="true" visible="true" x="16" y="83" width="52" height="23">
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="2" name="公众号" label="公众号" enabled="true" visible="true" x="16" y="83" width="52" height="22"/>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="16" y="129" width="60" height="60"/>
            <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="382" y="128" width="16" height="16"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="88" y="126" width="310" height="23">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="limi里美" name="limi里美" label="limi里美" enabled="true" visible="true" x="88" y="126" width="61" height="22"/>
            </XCUIElementTypeOther>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="88" y="151" width="310" height="21">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="真实好成分，呵护少女肌。" name="真实好成分，呵护少女肌。" label="真实好成分，呵护少女肌。" enabled="true" visible="true" x="88" y="152" width="184" height="19"/>
            </XCUIElementTypeOther>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="88" y="178" width="310" height="19">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="微信"

```

```

号: " name="微信号: " label="微信号: " enabled="true" visible="true" x="88" y="178" width="5
8" height="18"/>
    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="lim
i_official" name="limi_official" label="limi_official" enabled="true" visible="true" x="14
5" y="178" width="72" height="18"/>
    </XCUIElementTypeOther>
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible=
false" x="0" y="0" width="0" height="0"/>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible=
true" x="0" y="237" width="414" height="25">
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="没有
更多的搜索结果" name="没有更多的搜索结果" label="没有更多的搜索结果" enabled="true" visible="tr
ue" x="133" y="239" width="148" height="20"/>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    """
foundAccountId = soup.find(
    'XCUIElementTypeStaticText',
    attrs {"value": curAccountId, "name": curAccountId, "type": "XCUIElementTypeStatic
Text"}, )
logging.debug("foundAccountId=%s", foundAccountId)
# foundAccountId=<XCUIElementTypeStaticText enabled="true" height="18" label="gh_cfcfc
ee032cc" name="gh_cfcfce032cc" type="XCUIElementTypeStaticText" value="gh_cfcfce032cc" v
isible="true" width="112" x="145" y="178"/>
if foundAccountId:
    idParent = foundAccountId.parent
    logging.debug("idParent=%s", idParent)
    if idParent:
        # method 1: two prev.prev
        # # idParentPrev = idParent.previous_sibling
        # idParentPrev = idParent.previous_sibling.previous_sibling
        # accountDescNode = idParentPrev
        # logging.info("accountDescNode=%s", accountDescNode) # '\n'
        # if accountDescNode:
        #     # accountZhcnNode = accountDescNode.previous_sibling
        #     accountZhcnNode = accountDescNode.previous_sibling.previous_sibling
        #     logging.info("accountZhcnNode=%s", accountZhcnNode)

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

        # accountDescNode = None
        # accountZhcnNode = None

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

        # for eachPrevSiblingNode in idParentPrevSiblingList:
        #     curNodeName = eachPrevSiblingNode.name
        #     isTypeOtherNode = curNodeName == TypeOther

```

```

#     if isTypeOtherNode:
#         typeOtherNodeCurIdx += 1

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

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

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

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

#     )

# method 3: parent.parent is 搜一搜, direct child 2nd XCUIElementTypeOther of e
nabled="true" visible="true"
idParentParent = idParent.parent
if idParentParent:
    otherSoupList = idParentParent.find_all(
        "XCUIElementTypeOther",
        attrs = {"type": "XCUIElementTypeOther", "enabled":"true", "visible":"tr
ue"},

        recursive=False,
    )
    if otherSoupList and (len(otherSoupList) >= 2):
        firstOtherSoup = otherSoupList[0]
        if firstOtherSoup.attrs["name"] == "公众号":
            secondOtherSoup = otherSoupList[1]
            zhcnNameSoupList = secondOtherSoup.find_all(
                "XCUIElementTypeStaticText",
                attrs = {"type": "XCUIElementTypeStaticText", "enabled":"true",
"visible":"true"},

            )
            if zhcnNameSoupList:
                for eachTextSoup in zhcnNameSoupList:
                    curPartName = eachTextSoup.attrs.get("value")
                    accountZhcnFullName += curPartName

                if accountZhcnFullName:
                    secondOtherAttrDict = secondOtherSoup.attrs
                    parentX = secondOtherAttrDict["x"]
                    parentY = secondOtherAttrDict["y"]
                    parentWidth = secondOtherAttrDict["width"]
                    parentHeight = secondOtherAttrDict["height"]
                    parentNodeLocator = {
                        "type": "XCUIElementTypeOther",

```

```
        "enabled": "true",
        "visible": "true",
        "x": parentX,
        "y": parentY,
        "width": parentWidth,
        "height": parentHeight,
    }

# return accountZhcnTextSoup
# return accountZhcnFullName
return accountZhcnFullName, parentNodeLocator
```

调用举例：

```
curAccountId = "gh_cfcfcfcee032cc"
curPageXml = self.get_page_source()
soup = CommonUtils.xmlToSoup(curPageXml)

accountZhcnName, parentNodeLocator = self.findWeixinPublicAccountZhcnFullName(soup, curAccountId)
```

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2020-08-09 10:13:38

设置

此处整理iOS的内置app, bundle id是 com.apple.Preferences 的 设置 相关的通用代码段。

启动内置app: 设置

```

def iOSLaunchSettings(self, isTerminateFirst=True, isDebugState=True):
    """iOS terminal and re-launch Settings=Preferences app

    Args:
        isTerminateFirst (bool): force terminal Settings before launch
        isDebugState (bool): enable debug for app state
    Returns:
        bool
    Raises:
        """
    iOS_AppId_Settings = "com.apple.Preferences"

    if isTerminateFirst:
        if isDebugState:
            isGetOk, curState = self.iOSGetAppState(iOS_AppId_Settings)
            logging.info("before terminal: curState=%s", curState)
            # stop before start to avoid current page is not homepage of 设置
            isTerminalOk, respInfo = self.iOSTerminateApp(iOS_AppId_Settings)
            logging.info("%s: isTerminalOk=%s, respInfo=%s", iOS_AppId_Settings, isTerminalOk,
respInfo)
        if isDebugState:
            isGetOk, curState = self.iOSGetAppState(iOS_AppId_Settings)
            logging.info("after terminal: curState=%s", curState)

    # settingsSession = self.wdaClient.session(iOS_AppId_Settings)
    # logging.debug("settingsSession=%s" % settingsSession)
    # launchResult = self.wdaClient.app_launch(iOS_AppId_Settings)
    # logging.debug("launchResult=%s", launchResult)
    isLaunchOk, respInfo = self.iOSLaunchApp(iOS_AppId_Settings)
    logging.info("isLaunchOk=%s, respInfo=%s", isLaunchOk, respInfo)
    # logging.info("launchResult: value=%s, status=%s, sessionId=%s", launchResult.value,
    launchResult.status, launchResult.sessionId)
    # launchResult: value=None, status=0, sessionId=79A39B72-F5F9-4A01-8E58-DD380452350A
    # logging.info("launchResult=%s", str(launchResult))
    # launchResult=GenericDict(value=None, sessionId='79A39B72-F5F9-4A01-8E58-DD380452350A
    ', status=0)
    if isDebugState:
        isGetOk, curState = self.iOSGetAppState(iOS_AppId_Settings)
        logging.info("after launch: curState=%s", curState)
    return isLaunchOk, respInfo

```

调用:

```
isLaunchOk, respInfo = self.iOSLaunchSettings()
```

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09 10:13:38

更新WiFi代理

此处整理用 `facebook-wad` 自动实现关闭和恢复代理的整个自动化过程。

此处把

【已解决】iOS自动化安装app：给当前WiFi去掉代理以及自动安装app后再恢复之前代理中：

- 关闭代理
 - 给当前WiFi去掉代理
- 恢复代理
 - 给当前WiFi 加上代理

贴出来相关的逻辑和代理，供了解：iOS的自动化，是什么样的，代码大概怎么写流程和相关代码：

```
newProxyInfo = {
    "type": "关闭",
    "value": None,
}
isUpdateOk, respInfo = self.iOSWifiDetailUpdateProxy(newProxyInfo)
```

用于去关闭代理

(自动通过AppStore去安装iOS的app后) 后续去恢复 (之前保存的) 代理：

```
# # for debug
# # 'authUser': 'user', 'authPassword': '***'
# proxyConfigInfo = {
#     'type': '手动',
#     'value': {
#         'server': '192.168.31.46',
#         'port': '8081',
#         '# 'authenticate': '0'
#         'authenticate': '1',
#         'authUser': 'user',
#         'authPassword': 'pwd',
#     }
# }

# restore proxy if necessary
if proxyConfigInfo:
    # isRestoreOk = self.settingsRestoreWiFiProxy(proxyConfigInfo)
    isRestoreOk, respInfo = self.iOSWifiDetailUpdateProxy(proxyConfigInfo)
    logStr = "%s to restore previous proxy %s" % (isRestoreOk, proxyConfigInfo)
    if isRestoreOk:
        logging.info(logStr)
```

```

else:
    logging.error(logStr)

```

具体实现：

```

def iOSWifiDetailUpdateProxy(self, newProxyInfo):
    """iOS launch Settings, into WiFi list page, find current WiFi, into WiFi detail page
    then try update to new proxy config info
    Args:
        newProxyInfo (dict): new proxy config info
    Returns:
        bool, dict/str
            True, old proxy config info dict
            False, error message str
    Raises:
    """
    isUpdateProxyOk = False
    respInfo = None

    isGetProxyTypeOk, respInfo = self.iOSLaunchSettingsAndGetProxyType()
    if not isGetProxyTypeOk:
        respInfo = "Not find current WiFi proxy config type"
        return isUpdateProxyOk, respInfo

    curProxySoup = respInfo
    curProxyAttrDict = curProxySoup.attrs
    curTypeName = curProxyAttrDict.get("value")

    newTypeName = newProxyInfo["type"]
    if (newTypeName == "关闭") and (curTypeName == "关闭"):
        isUpdateProxyOk = True
        oldProxyInfo = {
            "type": "关闭",
            "value": None,
        }
        respInfo = oldProxyInfo
        return isUpdateProxyOk, respInfo
    else:
        # into config proxy page
        self.clickElementCenterPosition(curProxySoup)
        # get old proxy value
        # update to close
        # save
        isUpdateProxyOk, respInfo = self.iOSProxyConfigUpdateProxy(newProxyInfo)
        logging.info("Update proxy result: %s, %s", isUpdateProxyOk, respInfo)
        return isUpdateProxyOk, respInfo

```

其逻辑是：

启动 `设置`，直到进入当前已连接的WiFi的详情页，获取到当前代理的类型

其具体实现是：

```

def iOSLaunchSettingsAndGetProxyType(self):
    """
        iOS launch Settings, and into WiFi list page, click into current connected WiFi de
        tail page, find current proxy type
    Args:
    Returns:
        bool, soup/str
            True, proxy type soup
            False, error message str
    Raises:
    """
    isGetTypeOk = False
    respInfo = None

    isIntoWiFiListOk, respInfo = self.iOSLaunchSettingsAndIntoWiFiList()
    if not isIntoWiFiListOk:
        errMsg = respInfo
        respInfo = "Fail into WiFi list page for %s" % errMsg
        return isGetTypeOk, respInfo

    isIntoDetailOk = self.iOSFromWiFiListIntoWifiDetail()
    if not isIntoDetailOk:
        respInfo = "Fail go into WiFi detail page"
        return isGetTypeOk, respInfo

    proxyTypeSoup = self.iOSGetCurrentWiFiProxyType()
    if proxyTypeSoup:
        isGetTypeOk = True
        respInfo = proxyTypeSoup

    return isGetTypeOk, respInfo

```

即：

- 先启动 设置
- 再进去 WiFi 的列表页
- 再进去当前已连接 WiFi 的详情页

具体实现：

```

def iOSLaunchSettingsAndIntoWiFiList(self):
    """
        iOS launch Settings, then into / (sometime) already in WiFi list page

    Args:
    Returns:
        bool, None/str
            True, None
            False, error message str
    Raises:
    """
    isIntoWiFiListOk = False

```

```

respInfo = None

isLaunchOk, respInfo = self.iOSLaunchSettings()
if not isLaunchOk:
    respInfo = "Fail to launch 设置"
    return isIntoWiFiListOk, respInfo

# Special: sometime already being in WiFi list page, so need first check it
isInWifiList = self.iOSIsInWiFiList()
if isInWifiList:
    isIntoWiFiListOk = True
    respInfo = None
    return isIntoWiFiListOk, respInfo
else:
    # foundAndClickedWifi = self.iOSFromSettingsIntoWifiList()
    foundAndClickedWifi = CommonUtils.multipleRetry({"functionCallback": self.iOSFromSettingsIntoWifiList})
    if not foundAndClickedWifi:
        respInfo = "Not find 无线局域网 in 设置"
        return isIntoWiFiListOk, respInfo

    isInWifiList = self.iOSIsInWiFiList()
    if isInWifiList:
        isIntoWiFiListOk = True
        respInfo = None
        return isIntoWiFiListOk, respInfo
    else:
        isIntoWiFiListOk = False
        respInfo = "Unknown Error"
        return isIntoWiFiListOk, respInfo

```

启动设置

```

def iOSLaunchSettings(self, isTerminateFirst=True, isDebugState=True):
    """iOS terminal and re-launch Settings=Preferences app

Args:
    isTerminateFirst (bool): force terminal Settings before launch
    isDebugState (bool): enable debug for app state
Returns:
    bool
Raises:
    """

```

```

iOS_AppId_Settings = "com.apple.Preferences"

if isTerminateFirst:
    if isDebugState:
        isGetOk, curState = self.iOSGetAppState(iOS_AppId_Settings)
        logging.info("before terminal: curState=%s", curState)
    # stop before start to avoid current page is not homepage of 设置
    isTerminalOk, respInfo = self.iOSTerminateApp(iOS_AppId_Settings)

```

```
    logging.info("%s: isTerminalOk=%s, respInfo=%s", iOS_AppId_Settings, isTerminalOk,
respInfo)
    if isDebugState:
        isGetOk, curState = self.iOSGetAppState(iOS_AppId_Settings)
        logging.info("after terminal: curState=%s", curState)

# settingsSession = self.wdaClient.session(iOS_AppId_Settings)
# logging.debug("settingsSession=%s" % settingsSession)
# launchResult = self.wdaClient.app_launch(iOS_AppId_Settings)
# logging.debug("launchResult=%s", launchResult)
isLaunchOk, respInfo = self.iOSLaunchApp(iOS_AppId_Settings)
logging.info("isLaunchOk=%s, respInfo=%s", isLaunchOk, respInfo)
# logging.info("launchResult: value=%s, status=%s, sessionId=%s", launchResult.value,
launchResult.status, launchResult.sessionId)
# launchResult: value=None, status=0, sessionId=79A39B72-F5F9-4A01-8E58-DD380452350A
# logging.info("launchResult=%s", str(launchResult))
# launchResult=GenericDict(value=None, sessionId='79A39B72-F5F9-4A01-8E58-DD380452350A
', status=0)
if isDebugState:
    isGetOk, curState = self.iOSGetAppState(iOS_AppId_Settings)
    logging.info("after launch: curState=%s", curState)
return isLaunchOk, respInfo
```

对应页面 [设置 首页](#):



进入 WiFi 列表页

在 WiFi 列表页中 找 无线局域网，即可以找到当前已连接的 WiFi

页面 WiFi 列表页：



```

"无线局域网" label="无线局域网" enabled="true" visible="true" x="0" y="312" width="414" height="45">
    <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="false" x="20" y="319" width="29" height="29"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="无线局域网" name="无线局域网" label="无线局域网" enabled="true" visible="true" x="64" y="325" width="87" height="21"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="64" y="356" width="350" height="1"/>
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="xxx_guest_5G" name="xxx_guest_5G" label="xxx_guest_5G" enabled="true" visible="true" x="252" y="325" width="124" height="21"/>
                    <XCUIElementTypeButton type="XCUIElementTypeButton" name="更多信息" label="更多信息" enabled="false" visible="true" x="386" y="328" width="8" height="13"/>
                </XCUIElementTypeCell>
            """
        # parentWifiCellClassChain = "/XCUIElementTypeCell[`name = '无线局域网' AND rect.x = 0 AND rect.width = %d`]" % self.X
        parentWifiCellClassChain = "/XCUIElementTypeCell[`name = '无线局域网' AND rect.x = 0 AND rect.width = %d`]" % self.X
        wifiTextQuery = {"type": "XCUIElementTypeStaticText", "name": "无线局域网", "enabled": "true"}
        wifiTextQuery["parent_class_chains"] = [parentWifiCellClassChain]
        foundAndClickedWifi = self.findAndClickElement(query wifiTextQuery, timeout 0.1)
        return foundAndClickedWifi
    
```

判断是否已进入WiFi列表页：

```

def iOSIsInWiFiList(self):
    """Check whether is in WiFi list page or not

    Args:
    Returns:
        bool
    Raises:
    """
    isFoundWifi = False
    """
    设置 WiFi列表页:
        <XCUIElementTypeNavigationBar type="XCUIElementTypeNavigationBar" name="无线局域网" enabled="true" visible="true" x="0" y="20" width="414" height="44">
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="设置" label="设置" enabled="true" visible="true" x="0" y="20" width="66" height="44"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" name="无线局域网" label="无线局域网" enabled="true" visible="true" x="163" y="31" width="88" height="21"/>
        </XCUIElementTypeNavigationBar>
    """
    wifiName = "无线局域网"
    parentNaviBarClassChain = "/XCUIElementTypeNavigationBar[`name = '%s' AND rect.x = 0 AND rect.width = %d`]" % (wifiName, self.X)
    wifiQuery = {"type": "XCUIElementTypeOther", "name": wifiName, "enabled": "true"}
    wifiQuery["parent_class_chains"] = [parentNaviBarClassChain]
    
```

```
isFoundWifi, respInfo = self.findElement(query wifiQuery, timeout 0.1)
return isFoundWifi
```

且用多次判断，防止单次的失败。

再去从WiFi列表页进入详情页：

```
def iOSFromWifiListIntoWifiDetail(self):
    """from WiFi list page, click more info button into WiFi detail page """
    isIntoDetailOk = False
    """
        设置 无线局域网 列表页:
            <XCUIElementTypeTable type="XCUIElementTypeTable" enabled="true" visible="true"
    " x="0" y="0" width="414" height="736">
                ...
                    <XCUIElementTypeCell type="XCUIElementTypeCell" name="xxx_guest_5G, 安全网络, 信号强度 3 格, 共 3 格" label="xxx_guest_5G, 安全网络, 信号强度 3 格, 共 3 格" enabled="true" visible="true" x="0" y="144" width="414" height="43">
                        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="xxx_guest_5G" name="xxx_guest_5G" label="xxx_guest_5G" enabled="true" visible="true" x="40" y="155" width="278" height="21"/>
                        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="186" width="414" height="1"/>
                        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="8" y="151" width="28" height="29">
                            <XCUIElementTypeImage type="XCUIElementTypeImage" name="UIPreferencesBlueCheck" enabled="true" visible="false" x="8" y="151" width="28" height="29"/>
                        </XCUIElementTypeOther>
                        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="15" y="187" width="245" height="1"/>
                            <XCUIElementTypeImage type="XCUIElementTypeImage" name="Lock" enabled="true" visible="false" x="326" y="159" width="8" height="12"/>
                            <XCUIElementTypeImage type="XCUIElementTypeImage" name="WifiBars3" enabled="true" visible="false" x="346" y="153" width="16" height="25"/>
                            <XCUIElementTypeButton type="XCUIElementTypeButton" name="更多信息" label="更多信息" enabled="true" visible="true" x="372" y="154" width="22" height="22"/>
                        </XCUIElementTypeCell>
        ...
    curPageXml = self.get_page_source()
    soup = CommonUtils.xmlToSoup(curPageXml)
    blueCheckChainList = [
        {
            "tag": "XCUIElementTypeCell",
            "attrs": {"enabled": "true", "visible": "true", "x": "0", "width": "%s" % self.X}
        },
        {
            "tag": "XCUIElementTypeOther",
            "attrs": {"enabled": "true", "visible": "true"}
        },
        {
            "tag": "XCUIElementTypeImage",
            # "attrs": {"enabled": "true", "visible": "true", "name": "UIPreferencesBlueCheck"}
    ]
```

```
        "attrs": {"enabled":"true", "name": "UIPreferencesBlueCheck"}  
    },  
]  
blueCheckSoup = CommonUtils.bsChainFind(soup, blueCheckChainList)  
if blueCheckSoup:  
    parentOtherSoup = blueCheckSoup.parent  
    parentCellSoup = parentOtherSoup.parent  
    moreInfoSoup = parentCellSoup.find(  
        'XCUIElementTypeButton',  
        attrs={"type": "XCUIElementTypeButton", "name": "更多信息", "enabled":"true", "  
visible":"true"},  
    )  
    if moreInfoSoup:  
        clickedOk = self.clickElementCenterPosition(moreInfoSoup)  
        isIntoDetailOk = clickedOk  
return isIntoDetailOk
```

页面 WiFi详情页：



再从当前WiFi详情页，找到代理的类型：

```
def iOSGetCurrentWiFiProxyType(self):  
    """from WiFi detail page, get current proxy config type soup"""  
    proxyTypeSoup = None  
    """  
    设置 无线局域网 详情页 代理 关闭:  
        <XCUIElementTypeTable type="XCUIElementTypeTable" enabled="true" visible="true"  
        " x="0" y="0" width="414" height="736">  
        . . .  
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"  
        " x="0" y="0" width="414" height="736">  
    """
```

```

        true" x="0" y="657" width="414" height="39">
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="HTTP 代理" name="HTTP 代理" label="HTTP 代理" enabled="true" visible="true" x="0" y="657" width="414" height="39"/>
            </XCUIElementTypeOther>
            <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="0" y="695" width="414" height="46">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="695" width="414" height="1"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="740" width="414" height="1"/>
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="配置代理" name="配置代理" label="配置代理" enabled="true" visible="true" x="20" y="708" width="70" height="21"/>
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="关闭" name="关闭" label="关闭" enabled="true" visible="true" x="341" y="708" width="35" height="21"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="更多信息" label="更多信息" enabled="false" visible="true" x="386" y="711" width="8" height="14"/>
            </XCUIElementTypeCell>

        设置 无线局域网 详情页 代理 手动:
        <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="0" y="695" width="414" height="46">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="695" width="414" height="1"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="740" width="414" height="1"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="配置代理" name="配置代理" label="配置代理" enabled="true" visible="true" x="20" y="708" width="70" height="21"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="手动" name="手动" label="手动" enabled="true" visible="true" x="341" y="708" width="35" height="21"/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="更多信息" label="更多信息" enabled="false" visible="true" x="386" y="711" width="8" height="14"/>
        </XCUIElementTypeCell>
        ***
        curPageXml = self.get_page_source()
        soup = CommonUtils.xmlToSoup(curPageXml)
        configProxyChainList = [
            {
                "tag": "XCUIElementTypeTable",
                "attrs": self.FullScreenAttrDict
            },
            {
                "tag": "XCUIElementTypeCell",
                "attrs": {"enabled": "true", "visible": "true", "x": "0", "width": "%s" % self.X}
            },
            {
                "tag": "XCUIElementTypeStaticText",
                "attrs": {"enabled": "true", "visible": "true", "value": "配置代理"}
            },
        ],
    
```

```

    ]
configProxySoup = CommonUtils.bsChainFind(soup, configProxyChainList)
if configProxySoup:
    parentCellSoup = configProxySoup.parent
    # proxyTypeP = re.compile("(手动)|(自动)|(关闭)")
    # proxyTypeSoup = parentCellSoup.find(
    #     'XCUIElementTypeStaticText',
    #     attrs={"type": "XCUIElementTypeStaticText", "value": proxyTypeP, "enabled": "true",
    "visible": "true"},
    # )
    proxyTypeSoup = self.iOSFindChildProxyType(parentCellSoup, isReturnSoup=True)

return proxyTypeSoup

```

此处返回是 str (类型的字符串名称) 或soup (类型的soup节点)

```

def iOSFindChildProxyType(self, parentCellSoup, isReturnSoup=False):
    """from parent cell soup, find child proxy type node / node's name

Args:
    parentCellSoup (soup): Beautifulsoup soup of parent XCUIElementTypeCell
    isReturnSoup (bool): return soup if true, otherwise return soup's name
Returns:
    str/soup:
        str: 手动/自动/关闭
        soup: soup node
Raises:
"""
# proxyTypeName = None
# some cases:
"""
    设置 无线局域网 详情页 代理 关闭:
        <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true"
x="0" y="695" width="414" height="46">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
true" x="0" y="695" width="414" height="1"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
false" x="0" y="740" width="414" height="1"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="配置代理
" name="配置代理" label="配置代理" enabled="true" visible="true" x="20" y="708" width="70"
height="21"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="关闭" na
me="关闭" label="关闭" enabled="true" visible="true" x="341" y="708" width="35" height="21"
/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="更多信息" label="
更多信息" enabled="false" visible="true" x="386" y="711" width="8" height="14"/>
        </XCUIElementTypeCell>

    设置 无线局域网 详情页 代理 手动:
        <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true"
x="0" y="695" width="414" height="46">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="

```

```

        true" x="0" y="695" width="414" height="1"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
false" x="0" y="740" width="414" height="1"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="配置代理
" name="配置代理" label="配置代理" enabled="true" visible="true" x="20" y="708" width="70" h
eight="21"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="手动" na
me="手动" label="手动" enabled="true" visible="true" x="341" y="708" width="35" height="21"
/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="更多信息" label="
更多信息" enabled="false" visible="true" x="386" y="711" width="8" height="14"/>
        </XCUIElementTypeCell>

    设置 无线局域网 配置代理 手动:
        <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true"
x="0" y="144" width="414" height="45">
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="手动" na
me="手动" label="手动" enabled="true" visible="true" x="20" y="144" width="350" height="45"
/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
true" x="20" y="188" width="394" height="1"/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="更多信息" label="
更多信息" enabled="false" visible="true" x="380" y="161" width="14" height="11"/>
        </XCUIElementTypeCell>

    设置 无线局域网 配置代理 关闭:
        <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true"
x="0" y="99" width="414" height="45">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
true" x="0" y="99" width="414" height="1"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="关闭" na
me="关闭" label="关闭" enabled="true" visible="true" x="20" y="99" width="350" height="45"
/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
true" x="20" y="143" width="394" height="1"/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="更多信息" label="
更多信息" enabled="false" visible="true" x="380" y="116" width="14" height="11"/>
        </XCUIElementTypeCell>

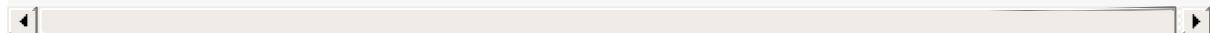
    设置 无线局域网 配置代理 自动:
        <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true"
x="0" y="189" width="414" height="45">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
true" x="0" y="233" width="414" height="1"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="自动" na
me="自动" label="自动" enabled="true" visible="true" x="20" y="189" width="350" height="45"
/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="更多信息" label="
更多信息" enabled="false" visible="true" x="380" y="206" width="14" height="11"/>
        </XCUIElementTypeCell>
        ***
proxyTypeP = re.compile("(手动)|(自动)|(关闭)")
proxyTypeSoup = parentCellSoup.find(

```

```

'XCUIElementTypeStaticText',
    attrs {"type": "XCUIElementTypeStaticText", "value": proxyTypeP, "enabled":"true",
"visible":"true"},
)
if isReturnSoup:
    return proxyTypeSoup
else:
    proxySoupAttrDict = proxyTypeSoup.attrs
    proxyTypeName = proxySoupAttrDict.get("value")
    return proxyTypeName # '手动'

```



从xml中找到 类型名

之后点击此soup节点

```

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

    Args:
        curElement (Element): Beautiful soup / lxml element / wda Element
    Returns:
        bool
    Raises:
    """

    hasClicked = False
    # centerPos = None
    centerX = None
    centerY = None

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

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

        if attrDict:

```

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

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

    return hasClicked
```

进入 配置代理 页面：



去更新代理:

```
def iOSProxyConfigUpdateProxy(self, newProxyInfo):
    """in proxy config = 代理配置 page, update proxy type

    Args:
        newProxyInfo (dict): new proxy info
    Returns:
        bool, dict/str
        True, old proxy config info
        False, error message str
```

```

Raises:
Examples:
    newProxyInfo examples:
        (1) to close:
        {
            "type": "关闭",
            "value": None
        }
        (2) to manual, no auth:
        {
            'type': '手动',
            'value': {
                'server': '192.168.31.47',
                'port': '8081',
                'authenticate': '0',
                'authUser': None,
                'authPassword': None
            }
        }
        (3) to manual, with auth:
        {
            'type': '手动',
            'value': {
                'server': '192.168.31.47',
                'port': '8081',
                'authenticate': '1'
                'authUser': 'user',
                'authPassword': 'password'
            }
        }
        (4) to auto:
        {
            'type': '自动',
            'value': {
                'url': 'your_auto_proxy_url'
            }
        }
    """
isUpdateOk = False
respInfo = None
"""

    设置 无线局域网 配置代理 关闭:
        <XCUIElementTypeTable type="XCUIElementTypeTable" enabled="true" visible="true"
        " x="0" y="0" width="414" height="736">
            <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true"
            " x="0" y="99" width="414" height="45">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
                " x="0" y="99" width="414" height="1"/>
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="关闭"
                " name="关闭" label="关闭" enabled="true" visible="true" x="20" y="99" width="350" height="45"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
                " x="20" y="143" width="394" height="1"/>

```

```

        <XCUIElementTypeButton type="XCUIElementTypeButton" name="更多信息" label="更多信息" enabled="false" visible="true" x="380" y="116" width="14" height="11"/>
        </XCUIElementTypeCell>
        <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="0" y="144" width="414" height="45">
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="手动" name="手动" label="手动" enabled="true" visible="true" x="20" y="144" width="374" height="45"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="20" y="188" width="394" height="1"/>
            </XCUIElementTypeCell>
            <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="0" y="189" width="414" height="45">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="233" width="414" height="1"/>
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="自动" name="自动" label="自动" enabled="true" visible="true" x="20" y="189" width="374" height="45"/>
            </XCUIElementTypeCell>
        </XCUIElementTypeTable>

```

设置 无线局域网 配置代理 手动:

```

<XCUIElementTypeTable type="XCUIElementTypeTable" enabled="true" visible="true" x="0" y="0" width="414" height="736">
    <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="0" y="99" width="414" height="45">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="99" width="414" height="1"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="关闭" name="关闭" label="关闭" enabled="true" visible="true" x="20" y="99" width="374" height="45"/>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="20" y="143" width="394" height="1"/>
        </XCUIElementTypeCell>
        <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="0" y="144" width="414" height="45">
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="手动" name="手动" label="手动" enabled="true" visible="true" x="20" y="144" width="350" height="45"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="20" y="188" width="394" height="1"/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="更多信息" label="更多信息" enabled="false" visible="true" x="380" y="161" width="14" height="11"/>
            </XCUIElementTypeCell>
            <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="0" y="189" width="414" height="45">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="233" width="414" height="1"/>
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="自动" name="自动" label="自动" enabled="true" visible="true" x="20" y="189" width="374" height="45"/>
            </XCUIElementTypeCell>
        </XCUIElementTypeTable>

```

```

        ue" x="0" y="269" width="414" height="46">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="269" width="414" height="1"/>
            <XCUIElementTypeTextField type="XCUIElementTypeTextField" value="192.1
68.31.46" name="服务器" label="服务器" enabled="true" visible="true" x="92" y="281" width="2
94" height="21"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="15" y="313" width="305" height="1"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="20" y="313" width="394" height="2"/>
        </XCUIElementTypeCell>
        <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="0" y="314" width="414" height="46">
            <XCUIElementTypeTextField type="XCUIElementTypeTextField" value="8081"
name="端口" label="端口" enabled="true" visible="true" x="74" y="326" width="312" height="22"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="15" y="358" width="305" height="2"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="20" y="359" width="394" height="1"/>
        </XCUIElementTypeCell>
        <XCUIElementTypeCell type="XCUIElementTypeCell" value="0" enabled="true" visible="true" x="0" y="359" width="414" height="46">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="404" width="414" height="1"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="鉴定"
name="鉴定" label="鉴定" enabled="true" visible="true" x="20" y="359" width="323" height="46"/>
            <XCUIElementTypeSwitch type="XCUIElementTypeSwitch" value="0" name="鉴定"
label="鉴定" enabled="true" visible="true" x="343" y="366" width="51" height="32"/>
        </XCUIElementTypeCell>
    </XCUIElementTypeTable>

```

设置 无线局域网 配置代理 自动:

```

    <XCUIElementTypeTable type="XCUIElementTypeTable" enabled="true" visible="true"
" x="0" y="0" width="414" height="736">
        <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true"
x="0" y="99" width="414" height="45">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false"
x="0" y="99" width="414" height="1"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="关闭"
name="关闭" label="关闭" enabled="true" visible="true" x="20" y="99" width="374" height="45"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
x="20" y="143" width="394" height="1"/>
        </XCUIElementTypeCell>
        <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true"
x="0" y="144" width="414" height="45">
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="手动"
name="手动" label="手动" enabled="true" visible="true" x="20" y="144" width="374" height="45"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
x="20" y="188" width="394" height="1"/>

```

```

        </XCUIElementTypeCell>
        <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="0" y="189" width="414" height="45">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="233" width="414" height="1"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="自动" name="自动" label="自动" enabled="true" visible="true" x="20" y="189" width="350" height="45"/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="更多信息" label="更多信息" enabled="false" visible="true" x="380" y="206" width="14" height="11"/>
        </XCUIElementTypeCell>
        <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="0" y="269" width="414" height="46">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="269" width="414" height="1"/>
            <XCUIElementTypeTextField type="XCUIElementTypeTextField" name="URL" label="URL" enabled="true" visible="true" x="72" y="281" width="314" height="21"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="313" width="414" height="2"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="15" y="313" width="305" height="1"/>
        </XCUIElementTypeCell>
    </XCUIElementTypeTable>
    """
# check current proxy type: 手动/自动/关闭
curPageXml = self.get_page_source()
soup = CommonUtils.xmlToSoup(curPageXml)
morenInfoChainList = [
    {
        "tag": "XCUIElementTypeTable",
        "attrs": self.FullScreenAttrDict
    },
    {
        "tag": "XCUIElementTypeCell",
        "attrs": {"enabled": "true", "visible": "true", "x": "0", "width": "%s" % self.X}
    },
    {
        "tag": "XCUIElementTypeButton",
        # "attrs": {"enabled": "false", "visible": "true", "name": "更多信息"}
        "attrs": {"visible": "true", "name": "更多信息"}
    },
]
morenInfoSoup = CommonUtils.bsChainFind(soup, morenInfoChainList)
if not morenInfoSoup:
    respInfo = "Fail to find 更多信息 in config proxy page"
    return isUpdateOk, respInfo

parentCellSoup = morenInfoSoup.parent
curTypeName = self.iOSFindChildProxyType(parentCellSoup)
if not curTypeName:
    respInfo = "Fail to find current proxy type name in config proxy page"
    return isUpdateOk, respInfo

```

```

parentTableSoup = parentCellSoup.parent

newTypeName = newProxyInfo["type"]
newProxyValue = newProxyInfo["value"]

curProxyValue = None
isSameType = False
isNeedSwitchType = True
isNeedGetCurValue = True
isNeedCompareValue = False
isNeedUpdatenewValue = True
isNeedStore = True

if newTypeName == curTypeName:
    isSameType = True
    isNeedSwitchType = False
    isNeedCompareValue = True
else:
    isNeedSwitchType = True

if curTypeName == "关闭":
    isNeedGetCurValue = False

if newTypeName == "关闭":
    isNeedUpdatenewValue = False

if (newTypeName == "关闭") and (curTypeName == "关闭"):
    isNeedStore = False
    isNeedCompareValue = False

if isNeedGetCurValue:
    if curTypeName == "手动":
        curProxyValue = self.getManualProxyValue(parentTableSoup)
    elif curTypeName == "自动":
        curProxyValue = self.getAutoProxyValue(parentTableSoup)

    if not curProxyValue:
        respInfo = "Fail to get %s proxy value" % curTypeName
        return isUpdateOk, respInfo

if isNeedCompareValue:
    # need check value is same or not
    if newProxyValue == curProxyValue:
        # if same, do nothing
        isNeedUpdatenewValue = False
        logging.info("No need change for same proxy type %s and value %s", curTypeName,
curProxyValue)

    # common logic process

if isNeedSwitchType:
    # switch to new type
    isSwitchOk = self.switchToProxyType(parentTableSoup, newTypeName)

```

```

if isSwitchOk:
    isNeedStore = True
else:
    respInfo = "Fail to switch to %s proxy" % curTypeName
    return isUpdateOk, respInfo

if isNeedUpdatenewValue:
    if newTypeName == "手动":
        isUpdateValueOk = self.setManualProxyValue(parentTableSoup, newProxyValue)
    elif newTypeName == "自动":
        isUpdateValueOk = self.setAutoProxyValue(parentTableSoup, newProxyValue)

    if isUpdateValueOk:
        isNeedStore = True
    else:
        respInfo = "Fail to update new %s proxy config value %s" % (curTypeName, newProxyValue)
        return isUpdateOk, respInfo

if isNeedStore:
    # type and/or value changed, need store
    isStoredOk = self.storeChangedProxyType()
    if not isStoredOk:
        respInfo = "Fail to store after proxy from %s switch to %s" % (curTypeName, newTypeName)
        return isUpdateOk, respInfo

isUpdateOk = True
oldProxyInfo = {
    "type": curTypeName,
    "value": curProxyValue,
}
return isUpdateOk, oldProxyInfo

```

实现了，复杂的逻辑处理。总体上支持如下全部各种状态之间互相切换：

- 关闭
- 自动
 - 没开 鉴定
 - 只有
 - 服务器
 - 端口
 - 鉴定=0
 - 开启 鉴定
 - 服务器
 - 端口
 - 鉴定=1
 - 有额外的
 - 用户名

- 密码
- 自动
 - 有
 - url值

且给出了传入的典型参数值：

(1) to close:

```
[  
    {  
        "type": "关闭",  
        "value": None  
    }  
]
```

(2) to manual, no auth:

```
[  
    {  
        'type': '手动',  
        'value': {  
            'server': '192.168.31.47',  
            'port': '8081',  
            'authenticate': '0',  
            'authUser': None,  
            'authPassword': None  
        }  
    }  
]
```

(3) to manual, with auth:

```
[  
    {  
        'type': '手动',  
        'value': {  
            'server': '192.168.31.47',  
            'port': '8081',  
            'authenticate': '1'  
            'authUser': 'user',  
            'authPassword': 'password'  
        }  
    }  
]
```

(4) to auto:

```
[  
    {  
        'type': '自动',  
        'value': {  
            'url': 'your_auto_proxy_url'  
        }  
    }  
]
```

其中内部调用到的函数是：

获取 手动 时的值

```

def getManualProxyValue(self, parentTableSoup):
    """in 配置代理 page, from parent table soup, find 手动 proxy value: server, port, authenticate

    Args:
        parentTableSoup (soup): parent table soup
    Returns:
        dict
    Raises:
        """
    manualProxyValue = None
    """
    设置 无线局域网 配置代理 手动:
        <XCUIElementTypeTable type="XCUIElementTypeTable" enabled="true" visible="true"
    " x="0" y="0" width="414" height="736">
            <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true"
    " x="0" y="99" width="414" height="45">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false"
    " x="0" y="99" width="414" height="1"/>
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="关闭"
    " name="关闭" label="关闭" enabled="true" visible="true" x="20" y="99" width="374" height="45"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
    " x="20" y="143" width="394" height="1"/>
            </XCUIElementTypeCell>
            <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true"
    " x="0" y="144" width="414" height="45">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="手动"
    " name="手动" label="手动" enabled="true" visible="true" x="20" y="144" width="350" height="45"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
    " x="20" y="188" width="394" height="1"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="更多信息" label="更多信息"
    " enabled="false" visible="true" x="380" y="161" width="14" height="11"/>
            </XCUIElementTypeCell>
            <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true"
    " x="0" y="189" width="414" height="45">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
    " x="0" y="233" width="414" height="1"/>
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="自动"
    " name="自动" label="自动" enabled="true" visible="true" x="20" y="189" width="374" height="45"/>
            </XCUIElementTypeCell>
            <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true"
    " x="0" y="269" width="414" height="46">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
    " x="0" y="269" width="414" height="1"/>
                <XCUIElementTypeTextField type="XCUIElementTypeTextField" value="192.1
    68.31.46" name="服务器" label="服务器" enabled="true" visible="true" x="92" y="281" width="2
    50" height="14"/>
        </XCUIElementTypeTable>
    
```

```

    94" height="21"/>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="15" y="313" width="305" height="1"/>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="20" y="313" width="394" height="2"/>
    </XCUIElementTypeCell>
    <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="0" y="314" width="414" height="46">
        <XCUIElementTypeTextField type="XCUIElementTypeTextField" value="8081" name="端口" label="端口" enabled="true" visible="true" x="74" y="326" width="312" height="22"/>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="15" y="358" width="305" height="2"/>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="20" y="359" width="394" height="1"/>
    </XCUIElementTypeCell>
    <XCUIElementTypeCell type="XCUIElementTypeCell" value="0" enabled="true" visible="true" x="0" y="359" width="414" height="46">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="404" width="414" height="1"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="鉴定" name="鉴定" label="鉴定" enabled="true" visible="true" x="20" y="359" width="323" height="46"/>
        <XCUIElementTypeSwitch type="XCUIElementTypeSwitch" value="0" name="鉴定" label="鉴定" enabled="true" visible="true" x="343" y="366" width="51" height="32"/>
    </XCUIElementTypeCell>
</XCUIElementTypeTable>
```

设置 无线局域网 配置代理 手动 开启 鉴定:

```

<XCUIElementTypeTable type="XCUIElementTypeTable" enabled="true" visible="true" x="0" y="0" width="414" height="736">
    <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="0" y="99" width="414" height="45">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="99" width="414" height="1"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="关闭" name="关闭" label="关闭" enabled="true" visible="true" x="20" y="99" width="374" height="45"/>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="20" y="143" width="394" height="1"/>
    </XCUIElementTypeCell>
    <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="0" y="144" width="414" height="45">
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="手动" name="手动" label="手动" enabled="true" visible="true" x="20" y="144" width="350" height="45"/>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="20" y="188" width="394" height="1"/>
        <XCUIElementTypeButton type="XCUIElementTypeButton" name="更多信息" label="更多信息" enabled="false" visible="true" x="380" y="161" width="14" height="11"/>
    </XCUIElementTypeCell>
    <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="0" y="189" width="414" height="45">
```

```

        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="233" width="414" height="1"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="自动" name="自动" label="自动" enabled="true" visible="true" x="20" y="189" width="374" height="45"/>
        </XCUIElementTypeCell>
        <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="0" y="269" width="414" height="46">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="269" width="414" height="1"/>
            <XCUIElementTypeTextField type="XCUIElementTypeTextField" value="192.168.31.46" name="服务器" label="服务器" enabled="true" visible="true" x="92" y="281" width="294" height="21"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="15" y="313" width="305" height="1"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="20" y="313" width="394" height="2"/>
        </XCUIElementTypeCell>
        <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="0" y="314" width="414" height="46">
            <XCUIElementTypeTextField type="XCUIElementTypeTextField" value="8081" name="端口" label="端口" enabled="true" visible="true" x="74" y="326" width="312" height="22"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="15" y="358" width="305" height="2"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="20" y="359" width="394" height="1"/>
        </XCUIElementTypeCell>
        <XCUIElementTypeCell type="XCUIElementTypeCell" value="1" enabled="true" visible="true" x="0" y="359" width="414" height="46">
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="鉴定" name="鉴定" label="鉴定" enabled="true" visible="true" x="20" y="359" width="323" height="46"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="20" y="404" width="394" height="1"/>
            <XCUIElementTypeSwitch type="XCUIElementTypeSwitch" value="1" name="鉴定" label="鉴定" enabled="true" visible="true" x="343" y="366" width="51" height="32"/>
        </XCUIElementTypeCell>
        <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="0" y="404" width="414" height="46">
            <XCUIElementTypeTextField type="XCUIElementTypeTextField" value="user" name="用户名" label="用户名" enabled="true" visible="true" x="92" y="416" width="294" height="22"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="15" y="448" width="305" height="2"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="20" y="449" width="394" height="1"/>
        </XCUIElementTypeCell>
        <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="0" y="449" width="414" height="47">
            <XCUIElementTypeSecureTextField type="XCUIElementTypeSecureTextField" value="•••" name="密码" label="密码" enabled="true" visible="true" x="74" y="461" width="312" height="22"/>

```

```

        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="494" width="414" height="2"/>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="15" y="493" width="305" height="2"/>
    </XCUIElementTypeCell>
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="495" width="414" height="31">
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="您的凭证可能以不加密方式发送" name="您的凭证可能以不加密方式发送" label="您的凭证可能以不加密方式发送" enabled="true" visible="true" x="0" y="495" width="414" height="31"/>
    </XCUIElementTypeOther>
    <XCUIElementTypeOther type="XCUIElementTypeOther" name="您的凭证可能以不加密方式发送" label="您的凭证可能以不加密方式发送" enabled="true" visible="false" x="0" y="495" width="414" height="31"/>
</XCUIElementTypeTable>
"""

# proxyServer = None
# proxyPort = None
# proxyAuthenticate = None
# <XCUIElementTypeTextField type="XCUIElementTypeTextField" value="192.168.31.46" name="服务器" label="服务器" enabled="true" visible="true" x="92" y="281" width="294" height="21"/>
proxyServerSoup = parentTableSoup.find(
    'XCUIElementTypeTextField',
    attrs {"type": "XCUIElementTypeTextField", "name": "服务器", "enabled": "true", "visible": "true"}, )
if not proxyServerSoup:
    return manualProxyValue
proxyServer = proxyServerSoup.attrs.get("value", None) # '192.168.31.46'

# <XCUIElementTypeTextField type="XCUIElementTypeTextField" value="8081" name="端口" label="端口" enabled="true" visible="true" x="74" y="326" width="312" height="22"/>
proxyPortSoup = parentTableSoup.find(
    'XCUIElementTypeTextField',
    attrs {"type": "XCUIElementTypeTextField", "name": "端口", "enabled": "true", "visible": "true"}, )
if not proxyPortSoup:
    return manualProxyValue
proxyPort = proxyPortSoup.attrs.get("value", None) # '8081'

# <XCUIElementTypeSwitch type="XCUIElementTypeSwitch" value="0" name="鉴定" label="鉴定" enabled="true" visible="true" x="343" y="366" width="51" height="32"/>
proxyAuthenticateSoup = parentTableSoup.find(
    'XCUIElementTypeSwitch',
    attrs {"type": "XCUIElementTypeSwitch", "name": "鉴定", "enabled": "true", "visible": "true"}, )
if not proxyAuthenticateSoup:
    return manualProxyValue
proxyAuthenticate = proxyAuthenticateSoup.attrs.get("value", None) # '0'

```

```

authUser = None
authPassword = None

if proxyAuthenticate == "1":
    # need save user and password
    # <XCUIElementTypeTextField type="XCUIElementTypeTextField" name="用户名" label="用
    户名" enabled="true" visible="true" x="92" y="416" width="294" height="22"/>
    authUserSoup = parentTableSoup.find(
        'XCUIElementTypeTextField',
        attrs={"type": "XCUIElementTypeTextField", "name": "用户名", "enabled": "true",
               "visible": "true"},
    )
    if not authUserSoup:
        return manualProxyValue
    authUser = authUserSoup.attrs.get("value", None) # 'user'

    # <XCUIElementTypeSecureTextField type="XCUIElementTypeSecureTextField" name="密码"
    label="密码" enabled="true" visible="true" x="74" y="461" width="312" height="22"/>
    authPasswordSoup = parentTableSoup.find(
        'XCUIElementTypeSecureTextField',
        attrs {"type": "XCUIElementTypeSecureTextField", "name": "密码", "enabled": "true",
               "visible": "true"},
    )
    if not authPasswordSoup:
        return manualProxyValue
    authPassword = authPasswordSoup.attrs.get("value", None) # '...'
    if '•' in authPassword:
        logging.warning("Get proxy authenticate password only get dot • -> Please user
self makesure the password is correct !")

manualProxyValue = {
    "server": proxyServer,
    "port": proxyPort,
    "authenticate": proxyAuthenticate,
    "authUser": authUser,
    "authPassword": authPassword,
}
logging.info("manualProxyValue=%s", manualProxyValue)
# manualProxyValue={'server': '192.168.31.46', 'port': '8081', 'authenticate': '0'}
# manualProxyValue={'server': '192.168.31.46', 'port': '8081', 'authenticate': '1', 'a
uthUser': 'user', 'authPassword': '...'}
return manualProxyValue

```

获取自动时的值：

页面：



```
def getAutoProxyValue(self, parentTableSoup):
    """in 配置代理 page, from parent table soup, find 自动 proxy value: url

    Args:
        parentTableSoup (soup): parent table soup
    Returns:
        dict
    Raises:
        """
    autoProxyValue = None
    """
```

```

    设置 无线局域网 配置代理 自动:
        <XCUIElementTypeTable type="XCUIElementTypeTable" enabled="true" visible="true"
        " x="0" y="0" width="414" height="736">
            <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true"
            " x="0" y="99" width="414" height="45">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false"
                " x="0" y="99" width="414" height="1"/>
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="关闭"
                " name="关闭" label="关闭" enabled="true" visible="true" x="20" y="99" width="374" height="45"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
                " x="20" y="143" width="394" height="1"/>
            </XCUIElementTypeCell>
            <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true"
            " x="0" y="144" width="414" height="45">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="手动"
                " name="手动" label="手动" enabled="true" visible="true" x="20" y="144" width="374" height="45"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
                " x="20" y="188" width="394" height="1"/>
            </XCUIElementTypeCell>
            <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true"
            " x="0" y="189" width="414" height="45">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
                " x="0" y="233" width="414" height="1"/>
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="自动"
                " name="自动" label="自动" enabled="true" visible="true" x="20" y="189" width="350" height="45"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="更多信息" label="更多信息"
                " enabled="false" visible="true" x="380" y="206" width="14" height="11"/>
            </XCUIElementTypeCell>
            <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true"
            " x="0" y="269" width="414" height="46">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
                " x="0" y="269" width="414" height="1"/>
                <XCUIElementTypeTextField type="XCUIElementTypeTextField" value="your_
auto_proxy_url" name="URL" label="URL" enabled="true" visible="true" x="72" y="281" width="314" height="21"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
                " x="0" y="313" width="414" height="2"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
                " x="15" y="313" width="305" height="1"/>
            </XCUIElementTypeCell>
        </XCUIElementTypeTable>
        """
        autoUrlSoup = parentTableSoup.find(
            'XCUIElementTypeTextField',
            attrs {"type": "XCUIElementTypeTextField", "name": "URL", "enabled": "true", "visible": "true"},
        )
        if not autoUrlSoup:
            return autoProxyValue
        autoProxyValue = autoUrlSoup.attrs.get("value", None) # 'your_auto_proxy_url'

```

```
    return autoProxyValue
```

点击去切换类型：

```
def switchToProxyType(self, parentTableSoup, newProxyTypeName):
    """in 配置代理 page, switch to new proxy type

    Args:
        parentTableSoup (soup): parent table soup
        newProxyTypeName (str): new proxy type name
    Returns:
        bool
    Raises:
        """
    isSwitchOk = False
    """
    设置 无线局域网 配置代理 手动:
    <XCUIElementTypeTable type="XCUIElementTypeTable" enabled="true" visible="true"
    " x="0" y="0" width="414" height="736">
        <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true"
        " x="0" y="99" width="414" height="45">
            ...
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="关闭"
            " name="关闭" label="关闭" enabled="true" visible="true" x="20" y="99" width="374" height="45"/>

            ...
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="手动"
            " name="手动" label="手动" enabled="true" visible="true" x="20" y="144" width="350" height="45"/>

            ...
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="自动"
            " name="自动" label="自动" enabled="true" visible="true" x="20" y="189" width="374" height="45"/>
    """

    newProxySoup = parentTableSoup.find(
        'XCUIElementTypeStaticText',
        attrs={"type": "XCUIElementTypeStaticText", "value": newProxyTypeName, "enabled": "true",
               "visible": "true"}, )
    if newProxySoup:
        clickedNewProxy = self.clickElementCenterPosition(newProxySoup)
        if clickedNewProxy:
            isSwitchOk = True
    return isSwitchOk
```

比如从 手动 切换到 关闭：



然后去恢复设置对应的值：

设置 手动 时的值：

```
def setManualProxyValue(self, parentTableSoup, newManualProxyValue):
    """in 配置代理 page, after changed to 手动
       set new manual proxy value

    Args:
        parentTableSoup (soup): parent table soup
        newManualProxyValue (dict): new manual proxy value dict
    Returns:
```

```

        bool
    Raises:
    """
    isUpdateManualOk = False
    """
        <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="0" y="269" width="414" height="46">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="269" width="414" height="1"/>
            <XCUIElementTypeTextField type="XCUIElementTypeTextField" value="192.168.31.46" name="服务器" label="服务器" enabled="true" visible="true" x="92" y="281" width="294" height="21"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="15" y="313" width="305" height="1"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="20" y="313" width="394" height="2"/>
            </XCUIElementTypeCell>
        """
    parentCellClassChain = "/XCUIElementTypeCell[`rect.x = 0 AND rect.width = %d`]" % self.x

    newServerValue = newManualProxyValue["server"]
    serverFieldQuery = {"type": "XCUIElementTypeTextField", "name": "服务器", "enabled": "true"}
    serverFieldQuery["parent_class_chains"] = [parentCellClassChain]
    # isFoundServer, respInfo = self.findElement(query=serverFieldQuery, timeout=0.1)
    # if not isFoundServer:
    #     return False
    isInputServerOk = self.wait_element_setText_iOS(serverFieldQuery, newServerValue)
    if not isInputServerOk:
        return False

    """
        <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="0" y="314" width="414" height="46">
            <XCUIElementTypeTextField type="XCUIElementTypeTextField" value="8081" name="端口" label="端口" enabled="true" visible="true" x="74" y="326" width="312" height="22"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="15" y="358" width="305" height="2"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="20" y="359" width="394" height="1"/>
            </XCUIElementTypeCell>
        """
    newPortValue = newManualProxyValue["port"]
    portFieldQuery = {"type": "XCUIElementTypeTextField", "name": "端口", "enabled": "true"}
    portFieldQuery["parent_class_chains"] = [parentCellClassChain]
    isInputPortOk = self.wait_element_setText_iOS(portFieldQuery, newPortValue)
    if not isInputPortOk:
        return False

    """
        <XCUIElementTypeCell type="XCUIElementTypeCell" value="0" enabled="true" v

```

```

    visible="true" x="0" y="359" width="414" height="46">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="404" width="414" height="1"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="鉴定" name="鉴定" label="鉴定" enabled="true" visible="true" x="28" y="359" width="323" height="46"/>
                <XCUIElementTypeSwitch type="XCUIElementTypeSwitch" value="0" name="鉴定" label="鉴定" enabled="true" visible="true" x="343" y="366" width="51" height="32"/>
            </XCUIElementTypeCell>
        ...
    newAuthenticateValue = newManualProxyValue["authenticate"]
    authSwitchQuery = {"type": "XCUIElementTypeSwitch", "name": "鉴定", "enabled": "true"}
    authSwitchQuery["parent_class_chains"] = [parentCellClassChain]
    foundAuth, respInfo = self.findElement(authSwitchQuery, timeout 0.1)
    if not foundAuth:
        return False

    authSwitchElement = respInfo

    curAuthValueStr = ""
    # curAuthValue = authSwitchElement.value # '0'
    # curAuthValueStr = str(curAuthValue)
    # Special: sometime wda element value is WRONG, actual is '1', but got '0'
    # so change to bs find then get value from page source xml
    curPageXml = self.get_page_source()
    soup = CommonUtils.xmlToSoup(curPageXml)
    authSwitchChainList = [
        {
            "tag": "XCUIElementTypeTable",
            "attrs": self.FullScreenAttrDict
        },
        {
            "tag": "XCUIElementTypeCell",
            "attrs": {"enabled": "true", "visible": "true", "x": "0", "width": "%s" % self.X}
        },
        {
            "tag": "XCUIElementTypeSwitch",
            "attrs": {"enabled": "true", "visible": "true", "name": "鉴定"}
        },
    ]
    authSwitchSoup = CommonUtils.bsChainFind(soup, authSwitchChainList)
    if authSwitchSoup:
        curAuthValue = authSwitchSoup.attrs.get("value", None)
        if curAuthValue:
            curAuthValueStr = str(curAuthValue)

    if curAuthValueStr == "":
        return False

    if curAuthValueStr != newAuthenticateValue:
        # click switch element to change value
        isClickOk = self.clickElement(authSwitchElement)
        if not isClickOk:

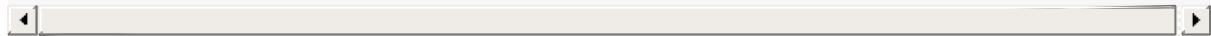
```

```
        return False

    if newAuthenticateValue == "1":
        # need restore auth user and password
        newAuthUserValue = newManualProxyValue["authUser"]
        userFieldQuery = {"type": "XCUIElementTypeTextField", "name": "用户名", "enabled": "true"}
        userFieldQuery["parent_class_chains"] = [ parentCellClassChain ]
        isInputUserOk = self.wait_element_setText_iOS(userFieldQuery, newAuthUserValue)
        if not isInputUserOk:
            return False

        newAuthPasswordValue = newManualProxyValue["authPassword"]
        passwordFieldQuery = {"type": "XCUIElementTypeSecureTextField", "name": "密码", "enabled": "true"}
        passwordFieldQuery["parent_class_chains"] = [ parentCellClassChain ]
        isInputPasswordOk = self.wait_element_setText_iOS(passwordFieldQuery, newAuthPasswordValue)
        if not isInputPasswordOk:
            return False

    isUpdateManualOk = True
    return isUpdateManualOk
```



页面：

支持 开启 鉴定 的值的恢复：



设置 自动 时的值：

```
def setAutoProxyValue(self, parentTableSoup, newAutoProxyValue):
    """in 配置代理 page, after changed to 自动
       set new manual proxy value
       by click each item then set value

    Args:
        parentTableSoup (soup): parent table soup
        newAutoProxyValue (dict): new auto proxy value dict
    Returns:
```

```

    bool
Raises:
"""
isUpdateAutoOk = False
"""
<XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="0" y="269" width="414" height="46">
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="269" width="414" height="1"/>
        <XCUIElementTypeTextField type="XCUIElementTypeTextField" value="your_auto_proxy_url" name="URL" label="URL" enabled="true" visible="true" x="72" y="281" width="314" height="21"/>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="313" width="414" height="2"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="15" y="313" width="305" height="1"/>
        </XCUIElementTypeCell>
"""
newUrlValue = newAutoProxyValue["url"]
parentCellClassChain = "/XCUIElementTypeCell[`rect.x = 0 AND rect.width = %d`]" % self.x
urlFieldQuery = {"type": "XCUIElementTypeTextField", "name": "URL", "enabled": "true"}
urlFieldQuery["parent_class_chains"] = [parentCellClassChain]
# foundUrl, respInfo = self.findElement(urlFieldQuery, timeout=0.1)
# if not foundUrl:
#     return False
isInputUrlOk = self.wait_element_setText_iOS(urlFieldQuery, newUrlValue)
isUpdateAutoOk = isInputUrlOk
return isUpdateAutoOk

```



当有变化后，右上角的存储

比如从 关闭 切换到 手动，且已填写完 手动时的配置后，右上角存储是蓝色 可以点击：



可以点击去保存改动：

```
def storeChangedProxyType(self):
    """in 配置代理 page, save changed proxy type, by click navigation bar 存储 button

Args:
    parentTableSoup (soup): parent table soup
    newProxyTypeName (str): new proxy type name
Returns:
    bool
Raises:
```

```
"""
isStoredOk = False
"""

设置 WiFi 配置代理 从手动切换到 关闭 存储:
<XCUIElementTypeNavigationBar type="XCUIElementTypeNavigationBar" name="配置代理" enabled="true" visible="true" x="0" y="20" width="414" height="44">
    <XCUIElementTypeButton type="XCUIElementTypeButton" name="xxx_guest_5G" label="xxx_guest_5G" enabled="true" visible="true" x="0" y="20" width="155" height="44"/>
    <XCUIElementTypeOther type="XCUIElementTypeOther" name="配置代理" label="配置代理" enabled="true" visible="true" x="172" y="31" width="70" height="21"/>
    <XCUIElementTypeButton type="XCUIElementTypeButton" name="存储" label="存储" enabled="true" visible="true" x="359" y="20" width="43" height="44"/>
</XCUIElementTypeNavigationBar>
"""

storeName = "存储"
parentNaviBarClassChain = "/XCUIElementTypeNavigationBar[`name = '配置代理' AND rect.x = 0 AND rect.width = %d`]" % self.X
storeButtonQuery = {"type": "XCUIElementTypeButton", "name": storeName, "enabled": "true"}
storeButtonQuery["parent_class_chains"] = [parentNaviBarClassChain]
foundAndClickedStore = self.findAndClickElement(query=storeButtonQuery, timeout=0.1)
isStoredOk = foundAndClickedStore
return isStoredOk
```

即可。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09 10:13:38

AppStore

此处整理用wda自动化操作AppStore的相关常见代码段。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09 10:13:38

自动安装app

此处整理用wda通过AppStore自动安装iOS的app的完整过程。

-» iPhone中用AppStore自动安装iOS的app的全过程

详见：

[【已解决】iOS自动抓包app：iPhone中通过AppStore自动安装iOS的app](#)

公共函数

下面函数中用到的公共函数，比如：

- `get_cmd_lines`
- `multipleRetry`
- `findElement`
- `findAndClickElement`

详见其他部分：

- [常用代码段](#)
- [元素处理](#)

过程

对应着的界面分别是：

启动AppStore后的，本身默认进入了Search的tab页：

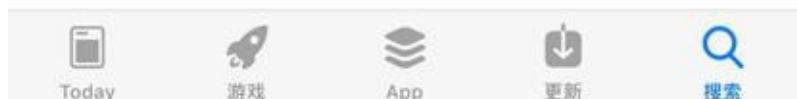
蜗牛移动

15:04

④ 4G

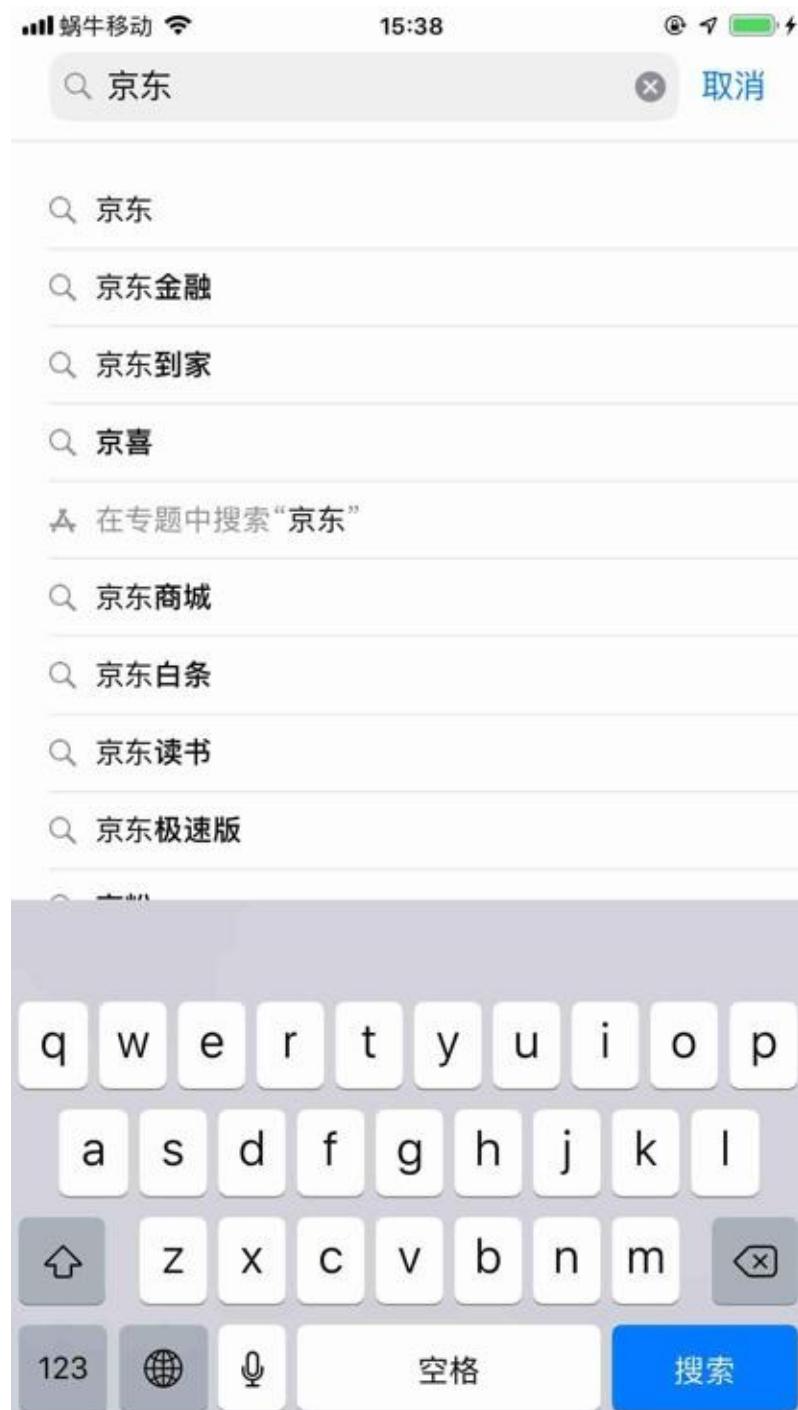
搜索

热门搜索

[个人所得税](#)[爱奇艺](#)[12306](#)[交管12123](#)[阿里巴巴](#)[京喜](#)[旅游](#)

当然，代码中防止不是默认搜索Tab，也用代码去切换到此tab页了。

然后再去点击搜索框，输入要搜索的app名字：



点击搜索后，进入搜索结果列表页：



然后找到列表页中第一个匹配的后，点击进入 app 下载详情页：



京东-一起热爱 就现在

新人送188元购物礼包



4.5 ★★★★★

34.6万个评分

#5

购物

17+

年龄

新功能

版本历史记录

版本 9.0.0

1 周前

京东9.0，全新启航 不负每一份热爱！618 惊喜剧透 火力全开！

5.29 超嗨预售抢先购

[更多](#)

预览



注意：

此处的图标是 云朵●中间有个 向下的箭头 表示 重新下载

意味着你的apple账号所登录过的iPhone中之前别处已经下载过该app了

如果是全新的没有下载过的app，则前面按钮显示的是获取



钉钉

阿里巴巴旗下移动办公平台

获取

...

2.5 ★★★★☆

213万个评分

#2

商务

4+

年龄



组织通讯录

人员关系一目了然，找人更



Today



游戏



App



更新



搜索

另外：

如果是付费的app，则显示的是金额：

蜗牛移动

16:19

④ ⌂

< 搜索



不折叠输入法

JUNJIE HUANG 行的小键盘

¥1.00

App 内购买项目

...

2.9 ★★★★☆

793个评分

#3

工具

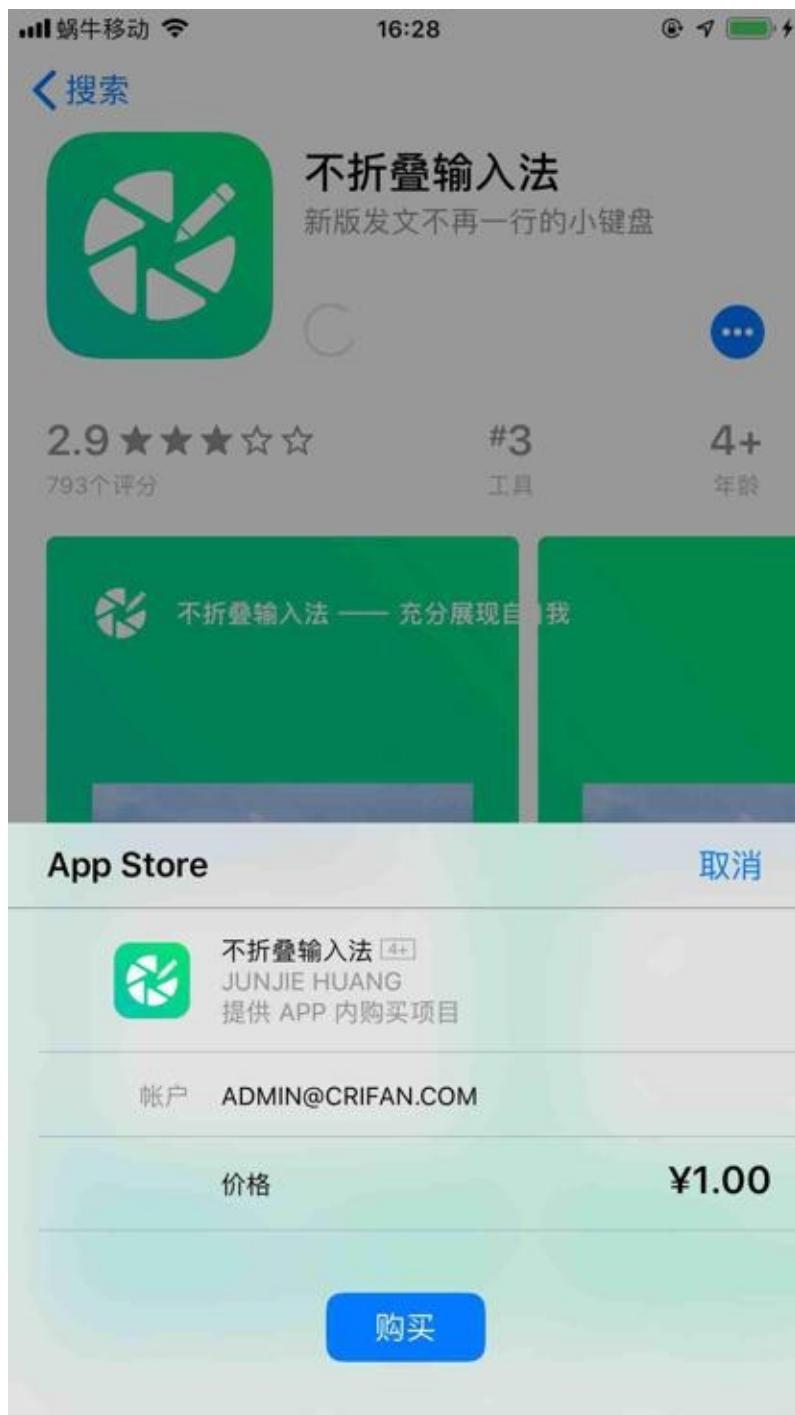
4+

年龄



对此：代码中，检测到是付费后，提示 不支持。

除非真的打算付费，否则点击后，弹框购买：

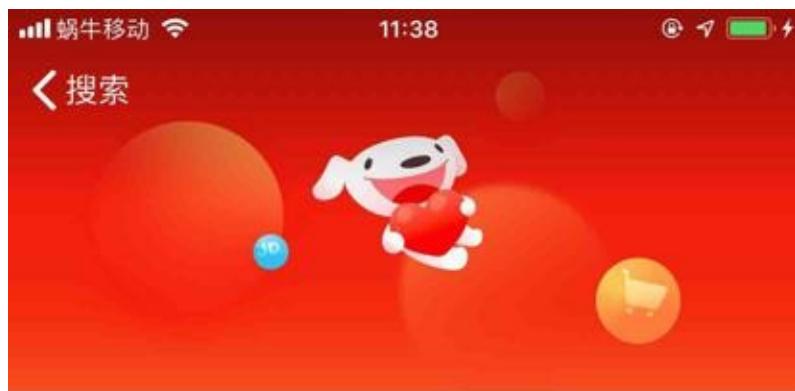


点击购买后，会真的扣费的。

点击下载按钮后：

情况1：对于（之前已下载过的app）重新下载，则无弹框

往往会出现加载中：



京东-一起热爱 就现在

新人送188元购物礼包



4.5 ★★★★★

35.4万个评分

#4

购物

17+

年龄

新功能

版本历史记录

版本 9.0.0

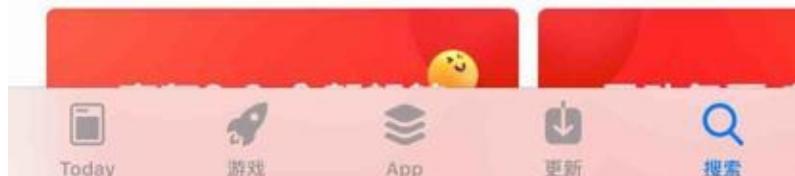
1 周前

京东9.0，全新启航 不负每一份热爱！618 惊喜剧透 火力全开！

5.29 超嗨预售抢先购

[更多](#)

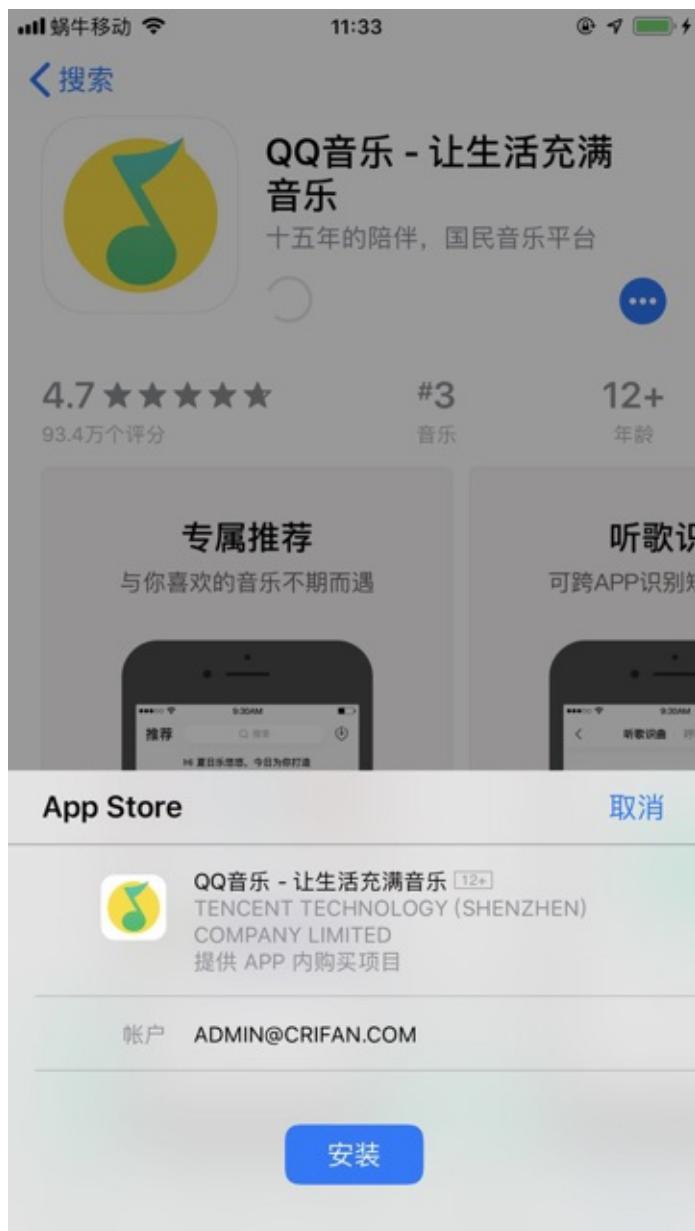
预览



情况2：对于（全新app）获取，会弹框安装

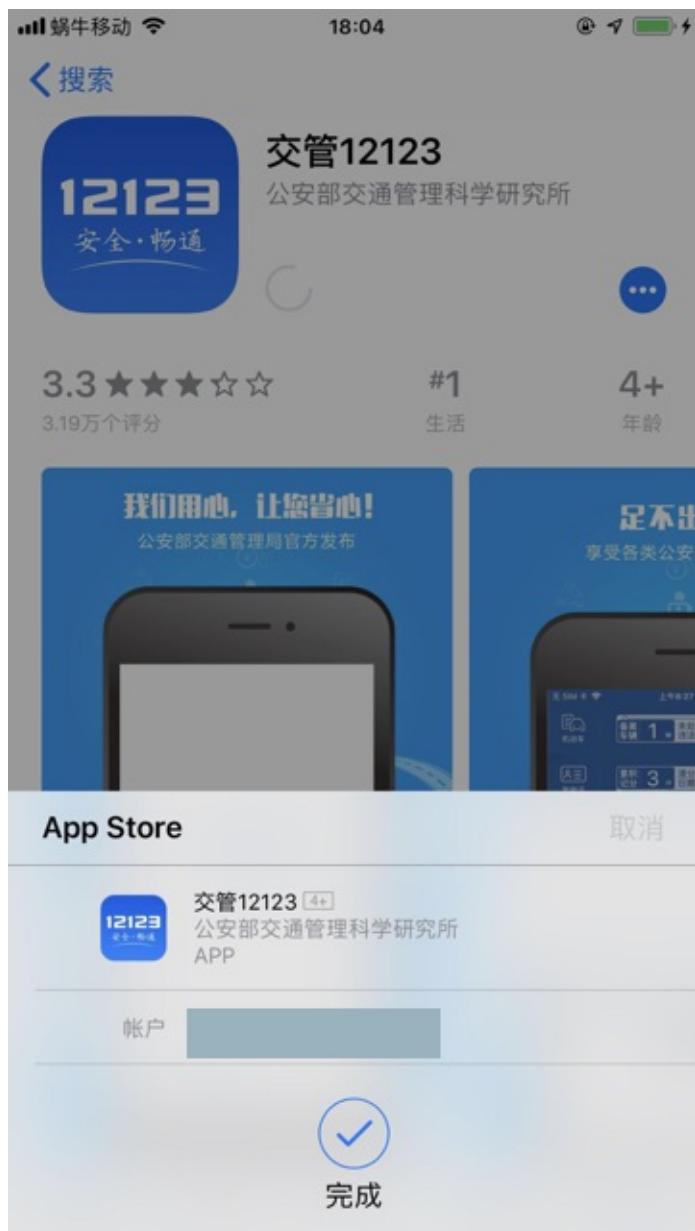
（借用别的app的弹框 展示）





点击安装，弹框会提示完成：

(其他app的截图)



之后就是下载过程了：

正在下载 其中圆圈 会有个蓝色进度条：



京东-一起热爱 就现在

新人送188元购物礼包

4.5 ★★★★★

35.4万个评分

#4

购物

17+

年龄

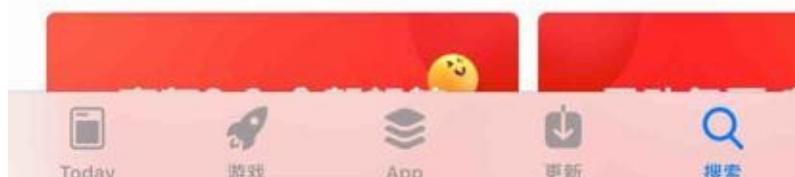
新功能**版本历史记录**

版本 9.0.0

1 周前

京东9.0，全新启航 不负每一份热爱！618 惊喜剧透 火力全开！

5.29 超嗨预售抢先购

[更多](#)**预览**

解释：

此处通过调试log日志：

```
[200608 14:22:43][DevicesMethods.py 1645] start get iOS page source
[200608 14:22:44][DevicesMethods.py 1658] Cost 1.57 seconds to get iOS page source
[200608 14:22:45][DevicesMethods.py 2238] Downloading 京东 progress: 0%
[200608 14:22:45][DevicesMethods.py 1645] start get iOS page source
[200608 14:22:45][DevicesMethods.py 1658] Cost 0.73 seconds to get iOS page source
[200608 14:22:45][DevicesMethods.py 2238] Downloading 京东 progress: 1%
[200608 14:22:45][DevicesMethods.py 1645] start get iOS page source
[200608 14:22:46][DevicesMethods.py 1658] Cost 0.74 seconds to get iOS page source
```

```
[200608 14:22:46][DevicesMethods.py 2238] Downloading 京东 progress: 4%
[200608 14:22:46][DevicesMethods.py 1645] start get iOS page source
[200608 14:22:47][DevicesMethods.py 1658] Cost 0.85 seconds to get iOS page source
[200608 14:22:47][DevicesMethods.py 2238] Downloading 京东 progress: 7%
[200608 14:22:47][DevicesMethods.py 1645] start get iOS page source
[200608 14:22:48][DevicesMethods.py 1658] Cost 0.80 seconds to get iOS page source
[200608 14:22:48][DevicesMethods.py 2238] Downloading 京东 progress: 9%
[200608 14:22:48][DevicesMethods.py 1645] start get iOS page source
[200608 14:22:48][DevicesMethods.py 1658] Cost 0.76 seconds to get iOS page source
[200608 14:22:48][DevicesMethods.py 2238] Downloading 京东 progress: 10%
[200608 14:22:48][DevicesMethods.py 1645] start get iOS page source
[200608 14:22:49][DevicesMethods.py 1658] Cost 0.78 seconds to get iOS page source
[200608 14:22:49][DevicesMethods.py 2238] Downloading 京东 progress: 14%
...
[200608 14:23:08][DevicesMethods.py 2238] Downloading 京东 progress: 60%
[200608 14:23:08][DevicesMethods.py 1645] start get iOS page source
[200608 14:23:09][DevicesMethods.py 1658] Cost 0.70 seconds to get iOS page source
[200608 14:23:09][DevicesMethods.py 2238] Downloading 京东 progress: 76%
```

多次调试发现：

在下载的进度超过76%之后，感觉内部就进入了自动安装过程

等安装完毕后，进度立刻就是100%，按钮变成后续的打开，表示安装完成了

-》即，推断是整体进度：

- 0~76%：下载进度
- 76%之后：安装进度
 - 但是不会显示，会从76%直接跳到打开
 - 表示安装完毕

最终安装完成后，显示 打开：



京东-一起热爱 就现在

新人送188元购物礼包

打开

...

4.5 ★★★★★

35.4万个评分

#4

购物

17+

年龄

新功能

版本历史记录

版本 9.0.0

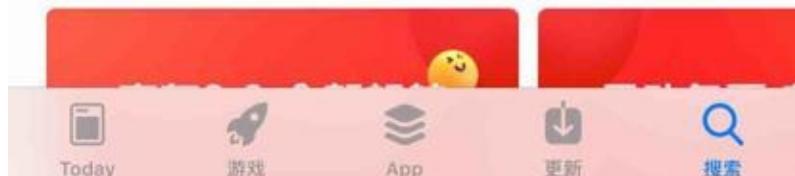
1 周前

京东9.0，全新启航 不负每一份热爱！618 惊喜剧透 火力全开！

5.29 超嗨预售抢先购

更多

预览



如果点击打开，即可启动app。

最后附上，部分测试后的iOS的app安装后桌面图标：



详见：

[【已解决】iOS自动化：通过AppStore自动下载和安装iOS的app京东](#)

代码

install_app_iOS 安装iOS的app

文件： src/common/DevicesMethods.py

```
def install_app_iOS(self, item, packages=None):
    """install iOS app
        if install ok, update bundleId for input item
        not install if found already installed
    """
    pass
```

```

isInstallOk = False
appInfo = None

# {'account': 'bb62512466_米家', 'bundleId': 'com.xiaomi.mihome', 'name': '米家', 'version': '4.30.2.0'}
appName = item["name"]

# for debug
# appName = "京东"
# appName = "斑马AI课"

# check if already installed
appInfo = self.getInstalledAppInfo(appName appName)
if appInfo:
    isInstallOk = True
    logging.warning("Not install %s for already installed %s", appName, appInfo)
else:
    # auto install app from AppStore
    isInstallOk, respInfo = self.iOSinstallAppFromAppStore(appName)
    logging.debug("appName=%s -> isInstallOk=%s, respInfo=%s", appName, isInstallOk, appInfo)
    # {'bundleId': 'com.fenbi.ape.zebstrika', 'name': '斑马AI课', 'version': '221'}
    if isInstallOk:
        appInfo = respInfo
    else:
        errMsg = respInfo
        logging.error("Fail to auto install iOS app %s, reason: %s", appName, errMsg)

if appInfo:
    # update bundle id
    installedBundleId = appInfo["bundleId"] # 'com.360buy.jdmobile', 'com.fenbi.ape.zebstrika'
    item["bundleId"] = installedBundleId
return isInstallOk

```

iOSinstallAppFromAppStore 从AppStore中自动安装iOS的app

```

def iOSinstallAppFromAppStore(self, appName):
    """Install iOS app from AppStore

Args:
    appName (str): app name
Returns:
    bool, dict/str
        bool: installed ok or not
        if true:
            dict: installed app info
        if false:
            str: fail reason error message
Raises:
"""
isInstallOk = False

```

```

respInfo = None

iOS_AppId_AppStore = "com.apple.AppStore"
appStoreSession = self.wdaClient.session(iOS_AppId_AppStore)
logging.debug("appStoreSession=%s" % appStoreSession)

isSwitchOk = CommonUtils.multipleRetry(
    {"functionCallback": self.switchToAppStoreSearchTab},
    maxRetryNum = 10,
    sleepInterval = 0.5,
)
if not isSwitchOk:
    respInfo = "Fail to switch to Search tab of AppStore"
    return isInstallOk, respInfo

"""
<XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="116" width="414" height="52">
    <XCUIElementTypeSearchField type="XCUIElementTypeSearchField" name="App Store" label="App Store" enabled="true" visible="true" x="20" y="117" width="374" height="36"/>
</XCUIElementTypeOther>
"""

searchInputQuery = {"type": "XCUIElementTypeSearchField", "name": "App Store"}
isInputOk = CommonUtils.multipleRetry(
    {
        "functionCallback": self.wait_element_setText,
        "functionParaDict": {
            "locator": searchInputQuery,
            "text": appName,
        }
    }
)
if not isInputOk:
    respInfo = "Fail to input app name %s into search field" % appName
    return isInstallOk, respInfo

isSearchOk = self.search_iOS(wait 0.2)
if not isSearchOk:
    respInfo = "Fail to find and click Search button to trigger search for %s" % appName
    return isInstallOk, respInfo

# Special: try add some wait time to avoid some special case:
# for 个人所得税 search result page, found and click 个人所得税, but sometime actually not into detail page
time.sleep(0.5)
isIntoDetailOk = CommonUtils.multipleRetry(
    {
        "functionCallback": self.appStoreSearchResultIntoDetail,
        "functionParaDict": {
            "appName": appName,
        }
    },
),

```

```

        maxRetryNum = 10,
        sleepInterval = 0.5,
    )
    if not isIntoDetailOk:
        respInfo = "Fail to into app detail page for %s" % appName
        return isInstallOk, respInfo
    # Special: try add some wait time to avoid some special case:
    # for 个人所得税 search result page, found and click 个人所得税, but sometime actually not into detail page
    time.sleep(0.2)

    detectRoundNum = 0
    while True:
        detectRoundNum += 1
        logging.info("%s try auto install, round [%d] %s", "-"*20, detectRoundNum, "-"*20)

        # foundOpen = False
        # isDownloading = False
        # isLoading = False
        # foundAndClickedPopupInstall = False
        # foundAndClickedDownload = False

        # isDownloading, curProgress = self.appStoreDownloadingProgress()
        # logging.info("isDownloading=%s, curProgress=%s", isDownloading, curProgress)
        isDownloading = self.appStoreDownloading()
        logging.info("isDownloading=%s", isDownloading)
        if isDownloading:
            # downloadingWaitTime = 0.5
            downloadingWaitTime = 1.0
            time.sleep(downloadingWaitTime)
            logging.info("Is downloading, wait %s seconds", downloadingWaitTime)
            continue

        foundOpen = self.appStoreCheckOpen()
        logging.info("foundOpen=%s", foundOpen)
        if foundOpen:
            logging.info("Found 打开 -> means %s is installed", appName)
            break

        isLoading = self.appStoreLoading()
        logging.info("isLoading=%s", isLoading)
        if isLoading:
            # loadingWaitTime = 0.2
            loadingWaitTime = 0.5
            time.sleep(loadingWaitTime)
            logging.info("Is loading, wait %s seconds", loadingWaitTime)
            continue

        foundAndClickedDownload, downloadButtonName = self.appStoreStartDownload(isShowErrMsgWhenNotFound=False)
        logging.info("foundAndClickedDownload=%s, downloadButtonName=%s", foundAndClickedDownload, downloadButtonName)
        if foundAndClickedDownload:

```

```

        logging.info("Found and clicked button %s to start download", downloadButtonName)
    else:
        foundMoneyButton, moneyButtonName = self.appStoreBuyAppMoneyButton()
        if foundMoneyButton:
            respInfo = "Not support auto install %s for need pay money %s" % (appName, moneyButtonName)
            return isInstallOk, respInfo

    foundAndClickedPopupInstall = self.appStoreClickPopupInstall()
    logging.info("foundAndClickedPopupInstall=%s", foundAndClickedPopupInstall)
    if foundAndClickedPopupInstall:
        popupInstallWaitTime = 0.2
        time.sleep(popupInstallWaitTime)
        logging.info("After click 安裝 of popup, wait %s seconds", popupInstallWaitTime)

    logging.info("Install complete for %s", appName)
    installedAppInfo = self.getInstalledAppInfo(appName)
    logging.info("installedAppInfo=%s", installedAppInfo)
    if not installedAppInfo:
        respInfo = "Fail to extract installed app info for %s" % appName
        return isInstallOk, respInfo

    isInstallOk = True
    respInfo = installedAppInfo
    logging.info("isInstallOk=%s, respInfo=%s", isInstallOk, respInfo)
    return isInstallOk, respInfo

```

其他相关函数：

getInstalledAppInfo 获取已安装的app的信息

```

def getInstalledAppInfo(self, appName=None, appBundleId=None):
    """find app info from app name or app bundle id

    Args:
        appName (str): iOS app name
        appBundleId (str): iOS app bundle id
    Returns:
        dict:
            app info
            eg: {'bundleId': 'com.360buy.jdmobile', 'name': '京东', 'version': '9.0.0'}
            None if not found
    Raises:
    """
    appInfo = None
    installedAppList = self.get_iOS_installedAppList()
    for eachAppInfo in installedAppList:
        eachAppName = eachAppInfo["name"]
        eachAppBundleId = eachAppInfo["bundleId"]

```

```

if appName:
    if eachAppName == appName:
        appInfo = eachAppInfo
        break

if appBundleId:
    if eachAppBundleId == appBundleId:
        appInfo = eachAppInfo
        break

return appInfo

```

get_iOS_installedAppList 获取已安装app的列表信息

```

def get_iOS_installedAppList(self):
    installedAppList = []
    listAppCmd = 'ideviceinstaller -l'
    appListStr = CommonUtils.get_cmd_lines(listAppCmd, text=True)
    logging.debug("appListStr=%s", appListStr)
    if appListStr:
        appRawList = appListStr.split("\n")
        """
        Total: 9 apps
        com.dianping.dpscope - 大众点评 10.27.10.21
        com.tencent.xin - 微信 7.0.12.33
        com.tencent.tiantianptu - 天天P图 603040
        com.didapinche.taxi - 嘻嗒出行 3
        com.luojilab.LuoJiFM-IOS - 得到 7.10.361
        com.suiyi.foodshop1 - 食行生鲜 49267
        com.alipay.iphoneclient - 支付宝 10.1.90.8000
        com.crifan.WebDriverAgentRunner.xctrunner - WebDriverAgentRunner-Runner 1
        com.xiaojukeji.didi - 滴滴出行 5.4.10.904142127
        """
        for eachAppStr in appRawList:
            eachAppStr = eachAppStr.strip()
            # foundApp = re.search("(?P<bundleId>com\.\S+)\s+-\s+(?P<name>\S+)\s+(?P<version>[\d\.\.]+)", eachAppStr)
            # rn.notes.best - 爱思极速版 11122019
            # foundApp = re.search("(?P<bundleId>[\w\.\.]+)\s+-\s+(?P<name>\S+)\s+(?P<version>[\d\.\.]+)", eachAppStr)
            # 'com.kingsoft.www.office.wpsoffice - WPS Office 10.11.076659'
            foundApp = re.search("(?P<bundleId>[\w\.\.]+)\s+-\s+(?P<name>[\S\ ]+)\s+(?P<version>[\d\.\.]+)", eachAppStr)
            if foundApp:
                bundleId = foundApp.group("bundleId") # 'com.dianping.dpscope'
                name = foundApp.group("name") # '大众点评'
                version = foundApp.group("version") # '10.27.10.21'
                curAppInfo = {
                    "bundleId": bundleId,
                    "name": name,
                    "version": version,
                }
                installedAppList.append(curAppInfo)

```

```

        installedAppList.append(curAppInfo)
    else:
        # Total: 9 apps
        if eachAppStr and (not eachAppStr.startswith("Total:")):
            logging.error("not match installed app item: %s", eachAppStr)
logging.debug("installedAppList=%s", installedAppList)
return installedAppList

```

switchToAppStoreSearchTab 切换到AppStore的Search的tab页

```

def switchToAppStoreSearchTab(self):
    """try find and click to switch to AppStore search tab"""
    isSwitchOk = False
    """
        AppStore 底部tab:
            <XCUIElementTypeTabBar type="XCUIElementTypeTabBar" enabled="true" visible="true" x="0" y="687" width="414" height="49">
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="Today" label="Today" enabled="true" visible="true" x="2" y="688" width="80" height="48"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="游戏" label="游戏" enabled="true" visible="true" x="85" y="688" width="80" height="48"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="App" label="App" enabled="true" visible="true" x="168" y="688" width="78" height="48"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="更新" label="更新" enabled="true" visible="true" x="249" y="688" width="80" height="48"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton" value="1" name="搜索" label="搜索" enabled="true" visible="true" x="332" y="688" width="80" height="48"/>
            </XCUIElementTypeTabBar>
    """
    parentTabBarClassChain = "/XCUIElementTypeTabBar[`rect.x = 0 AND rect.width = %d`]" % self.X
    searchButtonQuery = {"type": "XCUIElementTypeButton", "name": "搜索", "label": "搜索", "enabled": "true"}
    searchButtonQuery["parent_class_chains"] = [parentTabBarClassChain]
    foundAndClicked = self.findAndClickElement(query=searchButtonQuery, timeout=0.2)
    isSwitchOk = foundAndClicked
    return isSwitchOk

```

wait_element_setText_iOS 给元素输入值并等待一段时间

```

def wait_element_setText_iOS(self, query, text):
    isInputOk = False
    isFound, respInfo = self.findElement(query=query)
    logging.debug("isFound=%s, respInfo=%s", isFound, respInfo)
    if isFound:
        searchAccountElement = respInfo
        searchAccountElement.set_text(text)
        logging.info("has input text: %s", text)
        isInputOk = True
    return isInputOk

```

search_iOS iOS中点击弹出键盘中的Search触发搜索

```

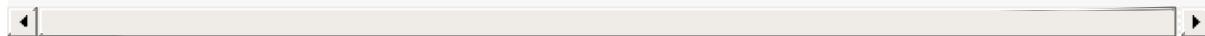
def search_iOS(self, wait=1):
    # 触发点击搜索按钮
    foundAndClickedDoSearch = False
    # <XCUIElementTypeButton type="XCUIElementTypeButton" name="Search" label="Search" enabled="true" visible="true" x="281" y="620" width="94" height="47"/>
    # <XCUIElementTypeButton type="XCUIElementTypeButton" name="Search" label="搜索" enabled="true" visible="true" x="309" y="685" width="103" height="50"/>
    searchButtonQuery = {"name": "Search"}
    searchButtonQuery = {"name": "Search", "type": "XCUIElementTypeButton"}
    # Note: occasionally not found Search, change to find multiple time to avoid this kind
    # of case
    MaxRetryNumber = 5
    curRetryNumber = MaxRetryNumber
    while curRetryNumber > 0:
        foundAndClickedDoSearch = self.findAndClickElement(searchButtonQuery, timeout=wait)

        if foundAndClickedDoSearch:
            break

        curRetryNumber -= 1

    if not foundAndClickedDoSearch:
        logging.error("Not found and/or clicked for %s", searchButtonQuery)
    return foundAndClickedDoSearch

```



appStoreSearchResultIntoDetail AppStore从搜索结果页中进去详情页

```

def appStoreSearchResultIntoDetail(self, appName):
    """for AppStore search result list page
       try find first match result
       then click into detail page

    Args:
        appName (str): app name
    Returns:
        bool, dict
            bool: is into detail page or not
    Raises:
        """
    isIntoDetailOk = False
    """
    搜索结果列表页 京东 重新下载:
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
        " x="0" y="0" width="414" height="736">

```



```

<XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="736">
    <XCUIElementTypeCollectionView type="XCUIElementTypeCollectionView" enabled="true" visible="true" x="0" y="0" width="414" height="736">
        <XCUIElementTypeCell type="XCUIElementTypeCell" name="京东-一起热爱
就现在，新人送188元购物礼包，四颗半星，34.6万 个评分" label="京东-一起热爱 就现在，新人送188元购
物礼包，四颗半星，34.6万 个评分" enabled="true" visible="true" x="20" y="86" width="374" hei
ght="295">
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="重新
下载" label="重新下载" enabled="true" visible="true" x="364" y="86" width="30" height="68"/>
        </XCUIElementTypeCell>
        <XCUIElementTypeCell type="XCUIElementTypeCell" name="京东金融-超过
三亿人在用，享24期免息，赚20亿金贴，四颗半星，20.8万 个评分" label="京东金融-超过三亿人在用，享24
期免息，赚20亿金贴，四颗半星，20.8万 个评分" enabled="true" visible="true" x="20" y="420" wid
th="374" height="295">
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="打开"
label="打开" enabled="true" visible="true" x="320" y="420" width="74" height="69"/>
        </XCUIElementTypeCell>
    </XCUIElementTypeCollectionView>
</XCUIElementTypeOther>
</XCUIElementTypeOther>

搜索结果列表页 美团 获取：
<XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
" x="0" y="0" width="414" height="736">
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
" x="0" y="0" width="414" height="736">
        <XCUIElementTypeCollectionView type="XCUIElementTypeCollectionView" en
abled="true" visible="true" x="0" y="0" width="414" height="736">
            <XCUIElementTypeCell type="XCUIElementTypeCell" name="美团-吃喝玩乐
尽在美团，美食外卖买菜买药首选美团App，四又四分之三颗星，94.7万 个评分" label="美团-吃喝玩乐 尽在
美团，美食外卖买菜买药首选美团App，四又四分之三颗星，94.7万 个评分" enabled="true" visible="tru
e" x="20" y="86" width="374" height="295">
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="获取"
label="获取" enabled="true" visible="true" x="320" y="86" width="74" height="68"/>
            </XCUIElementTypeCell>
            <XCUIElementTypeCell type="XCUIElementTypeCell" name="美团外卖-外卖订餐，送啥都快，美食生鲜，水果下午茶，四又四分之三颗星，607万 个评分" label="美团外卖-外卖订餐，送啥
都快，美食生鲜，水果下午茶，四又四分之三颗星，607万 个评分" enabled="true" visible="true" x="20"
y="420" width="374" height="295">
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="获取"
label="获取" enabled="true" visible="true" x="320" y="420" width="74" height="69"/>
            </XCUIElementTypeCell>
        </XCUIElementTypeCollectionView>
</XCUIElementTypeOther>
</XCUIElementTypeOther>

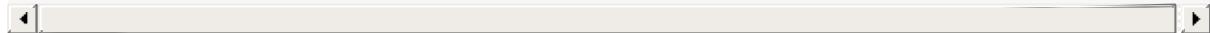
"""
parentCollectionViewClassChain = "/XCUIElementTypeCollectionView[`rect.x = 0 AND rect.
y = 0 AND rect.width = %d AND rect.height = %d`]" % (self.X, self.totally)
firstMatchCellQuery = {"type": "XCUIElementTypeCell", "nameContains": appName, "enabled"
: "true"}
firstMatchCellQuery["parent_class_chains"] = [parentCollectionViewClassChain]
foundAndClicked = self.findAndClickElement(query=firstMatchCellQuery, timeout=0.2)

```

```

isIntoDetailOk = foundAndClicked
return isIntoDetailOk

```



appStoreDownloading AppStore中是否是正在下载

```

# def appStoreDownloadingProgress(self):
def appStoreDownloading(self):
    """Detect app store is downloading some app

Args:
Returns:
    bool: true for found is downloading
Raises:
"""

isDownloading = False
# curProgress = ""
"""

AppStore 详情页 京东 正在下载:
<XCUIElementTypeCollectionView type="XCUIElementTypeCollectionView" enabled="true" visible="true" x="0" y="0" width="414" height="736">
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="-536" width="414" height="736"/>
    <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="0" y="200" width="414" height="231">
        <XCUIElementTypeImage type="XCUIElementTypeImage" value="京东-一起热爱就现在" name="插图" label="插图" enabled="true" visible="true" x="20" y="220" width="118" height="118"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="京东-一起热爱 就现在" name="京东-一起热爱 就现在" label="京东-一起热爱 就现在" enabled="true" visible="true" x="154" y="227" width="216" height="28"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="Beijing Jingdong Century Trading Co., Ltd." name="Beijing Jingdong Century Trading Co., Ltd." label="Beijing Jingdong Century Trading Co., Ltd." enabled="true" visible="false" x="154" y="257" width="232" height="39"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="新人送188元购物礼包" name="新人送188元购物礼包" label="新人送188元购物礼包" enabled="true" visible="true" x="154" y="257" width="148" height="19"/>
        <XCUIElementTypeButton type="XCUIElementTypeButton" value="18%" name="正在下载" label="正在下载" enabled="true" visible="true" x="154" y="308" width="74" height="30"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" enabled="true" visible="false" x="0" y="200" width="0" height="0"/>
        <XCUIElementTypeButton type="XCUIElementTypeButton" name="更多" label="更多" enabled="true" visible="true" x="366" y="310" width="28" height="28"/>
        <XCUIElementTypeOther type="XCUIElementTypeOther" name="四颗半星, 35.4万个评分, 4, 购物, 17+, 年龄" label="四颗半星, 35.4万个评分, 4, 购物, 17+, 年龄" enabled="true" visible="true" x="20" y="358" width="374" height="50"/>
    </XCUIElementTypeCell>
"""

# Note: here change wda query to bs.find, then revert back to use wda query

```

```

    # for later bs.find will need get page source, which too slow, and even sometime crash
    # or failed !

    parentCellClassChain = "/XCUIElementTypeCell[`rect.x = 0 AND rect.width = %d`]" % self.X
    downloadingButtonQuery = {"type": "XCUIElementTypeButton", "name": "正在下載", "enabled": "true"}
    downloadingButtonQuery["parent_class_chains"] = [parentCellClassChain]
    isfound, respInfo = self.findElement(query=downloadingButtonQuery, timeout=0.1)
    # if isfound:
    #     isDownloading = isfound
    #     curElement = respInfo
    #     curValue = curElement.value # always get null
    #     if curValue is not None:
    #         curProgress = curValue
    isDownloading = isfound

    # # Special: above wda query find element, get value, but got null
    # # so change to bs find
    # curPageXml = self.get_page_source()
    # soup = CommonUtils.xmlToSoup(curPageXml)
    isDownloadingChainList = [
        {
            "tag": "XCUIElementTypeCollectionView",
            "attrs": self.FullScreenAttrDict
        },
        {
            "tag": "XCUIElementTypeCell",
            "attrs": {"enabled": "true", "visible": "true", "x": "0", "width": "%s" % self.X}
        },
        {
            "tag": "XCUIElementTypeButton",
            "attrs": {"enabled": "true", "visible": "true", "name": "正在下載"}
        },
    ]
    isDownloadingSoup = CommonUtils.bsChainFind(soup, isDownloadingChainList)
    if isDownloadingSoup:
        isDownloading = True
        soupAttrDict = isDownloadingSoup.attrs
        curValue = soupAttrDict.get("value", "") # '0%'
        curProgress = curValue
    # return isDownloading, curProgress

    return isDownloading

```

appStoreCheckOpen 检测AppStore是否是下载完毕可以找到打开

```
def appStoreCheckOpen(self):
```

```

"""Detect whether app store is downloading complete then can found 打开

Args:
    Returns:
        bool
Raises:
"""

foundOpen = False
"""

AppStore 详情页 京东 打开:
<XCUIElementTypeCollectionView type="XCUIElementTypeCollectionView" enabled="true"
visible="true" x="0" y="0" width="414" height="736">
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
x="0" y="-536" width="414" height="736"/>
    <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true"
x="0" y="200" width="414" height="231">
        <XCUIElementTypeImage type="XCUIElementTypeImage" value="京东-一起热爱
就现在" name="插图" label="插图" enabled="true" visible="true" x="28" y="220" width="118" height="118"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="京东
-一起热爱 就现在" name="京东-一起热爱 就现在" label="京东-一起热爱 就现在" enabled="true" visible="true" x="154" y="227" width="216" height="28"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="Bei
jing Jingdong Century Trading Co., Ltd." name="Beijing Jingdong Century Trading Co., Ltd."
label="Beijing Jingdong Century Trading Co., Ltd." enabled="true" visible="false" x="154"
y="257" width="232" height="39"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="新人
送188元购物礼包" name="新人送188元购物礼包" label="新人送188元购物礼包" enabled="true" visible="true" x="154" y="257" width="148" height="19"/>
        <XCUIElementTypeButton type="XCUIElementTypeButton" name="打开" label="打开"
enabled="true" visible="true" x="154" y="308" width="74" height="30"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" enabled="true"
visible="false" x="0" y="200" width="0" height="0"/>
        <XCUIElementTypeButton type="XCUIElementTypeButton" name="更多" label="更多"
enabled="true" visible="true" x="366" y="310" width="28" height="28"/>
        <XCUIElementTypeOther type="XCUIElementTypeOther" name="四颗半星, 35.4
万个评分, 4, 购物, 17+, 年龄" label="四颗半星, 35.4万个评分, 4, 购物, 17+, 年龄" enabled="true"
visible="true" x="20" y="358" width="374" height="50"/>
    </XCUIElementTypeCell>
"""

parentCellClassChain = "/XCUIElementTypeCell[`rect.x = 0 AND rect.width = %d`]" % self.x
openButtonQuery = {"type": "XCUIElementTypeButton", "name": "打开", "enabled": "true"}
openButtonQuery["parent_class_chains"] = [parentCellClassChain]
foundOpen, respInfo = self.findElement(query=openButtonQuery, timeout=0.1)
return foundOpen

```

appStoreLoading 检测AppStore中是否是正在载入

```
def appStoreLoading(self):
```

```

"""after click start download button, check app store is loading or not

Args:
    Returns:
        bool
Raises:
"""

foundLoading = False
"""

AppStore 详情页 京东 正在载入:
<XCUIElementTypeCollectionView type="XCUIElementTypeCollectionView" enabled="true"
visible="true" x="0" y="0" width="414" height="736">
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
x="0" y="-536" width="414" height="736"/>
    <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true"
x="0" y="200" width="414" height="231">
        <XCUIElementTypeImage type="XCUIElementTypeImage" value="京东-一起热爱
就现在" name="插图" label="插图" enabled="true" visible="true" x="28" y="220" width="118"
height="118"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="京东
-一起热爱 就现在" name="京东-一起热爱 就现在" label="京东-一起热爱 就现在" enabled="true"
visible="true" x="154" y="227" width="216" height="28"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="Bei
jing Jingdong Century Trading Co., Ltd." name="Beijing Jingdong Century Trading Co., Ltd."
label="Beijing Jingdong Century Trading Co., Ltd." enabled="true" visible="false" x="154"
y="257" width="232" height="39"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="新人
送188元购物礼包" name="新人送188元购物礼包" label="新人送188元购物礼包" enabled="true"
visible="true" x="154" y="257" width="148" height="19"/>
        <XCUIElementTypeButton type="XCUIElementTypeButton" name="正在载入"
label="正在载入" enabled="true" visible="true" x="154" y="308" width="74" height="30"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" enabled="true"
visible="false" x="0" y="200" width="0" height="0"/>
        <XCUIElementTypeButton type="XCUIElementTypeButton" name="更多"
label="更多" enabled="true" visible="true" x="366" y="310" width="28" height="28"/>
        <XCUIElementTypeOther type="XCUIElementTypeOther" name="四颗半星, 35.4
万个评分, 4, 购物, 17+, 年龄" label="四颗半星, 35.4万个评分, 4, 购物, 17+, 年龄" enabled="true"
visible="true" x="20" y="358" width="374" height="50"/>
    </XCUIElementTypeCell>
"""

parentCellClassChain = "/XCUIElementTypeCell[`rect.x = 0 AND rect.width = %d`]" % self.
x
loadingButtonQuery = {"type": "XCUIElementTypeButton", "name": "正在载入", "enabled": "true"}
loadingButtonQuery["parent_class_chains"] = [parentCellClassChain]
foundLoading, respInfo = self.findElement(query=loadingButtonQuery, timeout=0.1)
return foundLoading

```

appStoreStartDownload 找到重新下载或获取等按钮并点击开始
下载

```

def appStoreStartDownload(self, isShowErrWhenNotFound=True):
    """being in app detail page inside AppStore
       try find and click download button

    Args:
        isShowErrWhenNotFound (bool): show error log when not found element
    Returns:
        bool, str:
            boo: found and clicked download button
            str: button name
    Raises:
        """
    # curPageXml = self.get_page_source()
    # soup = CommonUtils.xmlToSoup(curPageXml)
    foundAndClicked = False
    downloadButtonName = ""
    """
        AppStore 详情页 京东 重新下载:
        <XCUIElementTypeCollectionView type="XCUIElementTypeCollectionView" enabled="true" visible="true" x="0" y="0" width="414" height="736">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="-536" width="414" height="736"/>
                <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="0" y="200" width="414" height="231">
                    <XCUIElementTypeImage type="XCUIElementTypeImage" value="京东-一起热爱
就现在" name="插图" label="插图" enabled="true" visible="true" x="20" y="220" width="118" height="118"/>
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="京东
-一起热爱 就现在" name="京东-一起热爱 就现在" label="京东-一起热爱 就现在" enabled="true" visible="true" x="154" y="227" width="216" height="28"/>
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="Bei
jing Jingdong Century Trading Co., Ltd." name="Beijing Jingdong Century Trading Co., Ltd."
label="Beijing Jingdong Century Trading Co., Ltd." enabled="true" visible="false" x="154"
y="257" width="232" height="39"/>
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="新人
送188元购物礼包" name="新人送188元购物礼包" label="新人送188元购物礼包" enabled="true" visible
="true" x="154" y="257" width="148" height="19"/>
                    <XCUIElementTypeButton type="XCUIElementTypeButton" name="重新下载" lab
el="重新下载" enabled="true" visible="true" x="154" y="308" width="74" height="31"/>
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" enabled="t
rue" visible="false" x="0" y="200" width="0" height="0"/>
                    <XCUIElementTypeButton type="XCUIElementTypeButton" name="更多" label="
更多" enabled="true" visible="true" x="366" y="310" width="28" height="29"/>
                    <XCUIElementTypeOther type="XCUIElementTypeOther" name="四颗半星, 34.6
万个评分, 5, 购物, 17+, 年龄" label="四颗半星, 34.6万个评分, 5, 购物, 17+, 年龄" enabled="true
" visible="true" x="20" y="358" width="374" height="51"/>
                </XCUIElementTypeCell>
                <XCUIElementTypeOther type="XCUIElementTypeOther" name="新功能" label="新功
能" enabled="true" visible="true" x="0" y="430" width="414" height="49">
                    <XCUIElementTypeButton type="XCUIElementTypeButton" name="版本历史记录"
label="版本历史记录" enabled="true" visible="true" x="290" y="439" width="104" height="34"/>
                </XCUIElementTypeOther>
                <XCUIElementTypeCell type="XCUIElementTypeCell" name="版本 9.0.0, 1 周前,>
    
```

京东9.0，全新启航 不负每一份热爱！6.18 惊喜剧透 火力全开！5.29 超嗨预售抢先购 定金膨胀不止5倍 御兽刮刮乐解锁千万红包 5.30 超省百亿消费券 0.99元享超值券包 5.31超燃大势新品赏 大牌新品1元抢 6.1 开门红 狂撒超级百亿补贴 6.2 大牌爆品全聚惠 极致性价比更有24期免息 6.5 京东超级品牌日 全球好物 超级盛典 6.8 PLUS DAY Plus会员购物限时返20倍京豆 5.31-6.15 来京东 看爱豆 选大牌 瓜分3亿京豆 6.1-6.18 狂欢嗨潮 福利驾到 全品类接力庆生钜惠 各品牌携手感恩回馈 京东618十七周年庆 一起热爱就现在！感恩有你

最近更新： 【全新首页】首頁风格焕然一新，更灵动，更轻松 【家庭号】家人购物，贴心代付，专享家庭权益 【极速到家】买水买药，商超便利，半小时达 【排行榜】爆款单品，潮物推荐，跟榜购好物 【到手好价】折扣促销，极速支付，轻松享好价 【花样晒单】主题晒单，互动玩法，不止是购物" label="版本 9.0.0, 1 周前，京东9.0，全新启航 不负每一份热爱！6.18 惊喜剧透 火力全开！5.29 超嗨预售抢先购 定金膨胀不止5倍 御兽刮刮乐解锁千万红包 5.30 超省百亿消费券 0.99元享超值券包 5.31超燃大势新品赏 大牌新品1元抢 6.1 开门红 狂撒超级百亿补贴 6.2 大牌爆品全聚惠 极致性价比更有24期免息 6.5 京东超级品牌日 全球好物 超级盛典 6.8 PLUS DAY Plus会员购物限时返20倍京豆 5.31-6.15 来京东 看爱豆 选大牌 瓜分3亿京豆 6.1-6.18 狂欢嗨潮 福利驾到 全品类接力庆生钜惠 各品牌携手感恩回馈 京东618十七周年庆 一起热爱就现在！感恩有你

最近更新： 【全新首页】首頁风格焕然一新，更灵动，更轻松 【家庭号】家人购物，贴心代付，专享家庭权益 【极速到家】买水买药，商超便利，半小时达 【排行榜】爆款单品，潮物推荐，跟榜购好物 【到手好价】折扣促销，极速支付，轻松享好价 【花样晒单】主题晒单，互动玩法，不止是购物" enabled="true" visible="true" x="20" y="478" width="374" height="104"/>

AppStore 详情页 钉钉 获取：

```

<XCUIElementTypeCollectionView type="XCUIElementTypeCollectionView" enabled="true" visible="true" x="0" y="0" width="414" height="736">
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="-536" width="414" height="736"/>
        <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="0" y="200" width="414" height="231">
            <XCUIElementTypeImage type="XCUIElementTypeImage" value="钉钉" name="插图" label="插图" enabled="true" visible="true" x="20" y="220" width="118" height="118"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="钉钉" name="钉钉" label="钉钉" enabled="true" visible="true" x="154" y="227" width="45" height="28"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="DingTalk (China) Information Technology Co., Ltd." name="DingTalk (China) Information Technology Co., Ltd." label="DingTalk (China) Information Technology Co., Ltd." enabled="true" visible="false" x="154" y="257" width="199" height="39"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="阿里巴巴旗下移动办公平台" name="阿里巴巴旗下移动办公平台" label="阿里巴巴旗下移动办公平台" enabled="true" visible="true" x="154" y="257" width="184" height="19"/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="获取" label="获取" enabled="true" visible="true" x="154" y="308" width="74" height="31"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" enabled="true" visible="false" x="0" y="200" width="0" height="0"/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="更多" label="更多" enabled="true" visible="true" x="366" y="310" width="28" height="29"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" name="两颗半星，213万个评分，2，商务，4+，年龄" label="两颗半星，213万个评分，2，商务，4+，年龄" enabled="true" visible="true" x="20" y="358" width="374" height="51"/>
        </XCUIElementTypeCell>
    ***
    # downloadP = re.compile("(获取)|(重新下载)")
    # downloadChainList = [
    #     {
    #         "tag": "XCUIElementTypeCollectionView",
    #         "attrs": self.FullScreenAttrDict

```

```

#     },
#     {
#         "tag": "XCUIElementTypeCell",
#         "attrs": {"enabled":"true", "visible":"true", "x":"0", "width":"%s" % self.x}

#     },
#     {
#         "tag": "XCUIElementTypeButton",
#         "attrs": {"enabled":"true", "visible":"true", "name": downloadP}
#     },
# ]
# downloadSoup = CommonUtils.bsChainFind(soup, downloadChainList)
# if downloadSoup:
#     self.clickElementCenterPosition(downloadSoup)
#     foundAndClickedDownload = True
# return foundAndClickedDownload

# Note: to avoid possible later get page slow or even fail
# change to wda query
parentCellClassChain = "/XCUIElementTypeCell[`rect.x = 0 AND rect.width = %d`]" % self.
x
downloadButtonNameList = ["获取", "重新下载"]
for eachDownloadButtonName in downloadButtonNameList:
    curDownloadButtonQuery = {"type": "XCUIElementTypeButton", "name": eachDownloadButt
onName, "enabled": "true"}
    curDownloadButtonQuery["parent_class_chains"] = [parentCellClassChain]
    foundAndClicked = self.findAndClickElement(query=curDownloadButtonQuery, timeout=0
.1, isShowErrLog=isShowErrWhenNotFound)
    if foundAndClicked:
        downloadButtonName = eachDownloadButtonName
        break
return foundAndClicked, downloadButtonName

```



appStoreBuyAppMoneyButton 检测AppStore中是否有¥金额，
表示是收费应用

```

def appStoreBuyAppMoneyButton(self):
    """being in app detail page inside AppStore
       try find buy app button which has money text like ¥1.00

    Args:
    Returns:
        bool, str:
            boo: found and clicked download button
            str: button name
    Raises:
    """
    foundMoneyButton = False
    moneyButtonName = ""
    """

```

```

    AppStore 详情页 不折叠输入法 带金额的购买按钮 ¥1.00:
        <XCUIElementTypeCollectionView type="XCUIElementTypeCollectionView" enabled="true"
        visible="true" x="0" y="0" width="414" height="736">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false"
            x="0" y="-672" width="414" height="736"/>
            <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true"
            x="0" y="64" width="414" height="211">
                <XCUIElementTypeImage type="XCUIElementTypeImage" value="不折叠输入法" name="插图" label="插图" enabled="true" visible="true" x="20" y="64" width="118" height="118" />
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="不折叠输入法" name="不折叠输入法" label="不折叠输入法" enabled="true" visible="true" x="154" y="71" width="134" height="27"/>
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="JUN JIE HUANG" name="JUNJIE HUANG" label="JUNJIE HUANG" enabled="true" visible="false" x="154" y="101" width="108" height="19"/>
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="新版发文不再一行的小键盘" name="新版发文不再一行的小键盘" label="新版发文不再一行的小键盘" enabled="true" visible="true" x="154" y="101" width="184" height="19"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="¥1.00" label="¥1.00" enabled="true" visible="true" x="154" y="152" width="74" height="30"/>
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="App 内购买项目" name="App 内购买项目" label="App 内购买项目" enabled="true" visible="true" x="234" y="152" width="62" height="30"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="更多" label="更多" enabled="true" visible="true" x="366" y="154" width="28" height="28"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" name="三颗星, 793个评分, 3, 工具, 4+, 年龄" label="三颗星, 793个评分, 3, 工具, 4+, 年龄" enabled="true" visible="true" x="20" y="202" width="374" height="50"/>
            </XCUIElementTypeCell>
        """
        parentCellClassChain = "/XCUIElementTypeCell[`rect.x = 0 AND rect.width = %d`]" % self.x
        moneyButtonQuery = {"type": "XCUIElementTypeButton", "nameContains": "¥", "enabled": "true"}
        moneyButtonQuery["parent_class_chains"] = [parentCellClassChain]
        foundMoneyButton, respInfo = self.findElement(query=moneyButtonQuery, timeout=0.1)
        if foundMoneyButton:
            curElement = respInfo
            # internally use first non-empty of element.wdValue, element.wdLabel, not expected meaning
            # curName = curElement.text
            # so change to name, Note: has changed facebook-wad get name from self._prop('name') to self._prop('attribute/name')
            curName = curElement.name
            if curName is not None:
                moneyButtonName = curName
        return foundMoneyButton, moneyButtonName

```

appStoreClickPopupInstall AppStore中点击弹框开始安装

```

def appStoreClickPopupInstall(self):
    """after click start download button, try find popup install window then click install

    Args:
    Returns:
        bool
    Raises:
    """

    """
    AppStore 详情页 淘宝 弹框 安装:
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
    " x="0" y="515" width="414" height="221">
            <XCUIElementTypeTable type="XCUIElementTypeTable" enabled="true" visible="
    true" x="0" y="515" width="414" height="120">
                <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible
    ="true" x="0" y="515" width="414" height="77">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" v
    isible="true" x="16" y="590" width="398" height="1"/>
                    <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" v
    isible="false" x="55" y="533" width="40" height="41"/>
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value=
    "手机淘宝 - 淘到你说好 TAOBAO (CHINA) SOFTWARE CO.,LTD APP" name="手机淘宝 - 淘到你说好 TAOB
    AO (CHINA) SOFTWARE CO.,LTD APP" label="手机淘宝 - 淘到你说好 TAOBAO (CHINA) SOFTWARE CO.,L
    TD APP" enabled="true" visible="true" x="110" y="530" width="234" height="48"/>
                </XCUIElementTypeCell>
                <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible
    ="true" x="0" y="591" width="414" height="45">
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value=
    "帐户" name="帐户" label="帐户" enabled="true" visible="true" x="16" y="606" width="79" heig
    ht="16"/>
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value=
    "xxx@xxx.com" name="xxx@xxx.com" label="xxx@xxx.com" enabled="true" visible="true" x="111"
    y="606" width="134" height="16"/>
                </XCUIElementTypeCell>
            </XCUIElementTypeTable>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
    true" x="0" y="635" width="414" height="101">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visib
    le="true" x="16" y="635" width="398" height="1"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visib
    le="true" x="0" y="655" width="414" height="69"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="安装" label="
    安装" enabled="true" visible="true" x="165" y="667" width="84" height="37"/>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    """
    parentOtherClassChain = "/XCUIElementTypeOther[`rect.x = 0 AND rect.width = %d`]" % se
if X
    installButtonQuery = {"type": "XCUIElementTypeButton", "name": "安装", "enabled": "true"}

    installButtonQuery["parent_class_chains"] = [parentOtherClassChain]
    foundAndClickedInstall = self.findAndClickElement(query=installButtonQuery, timeout=0.1

```

```
, isShowErrLog False)
    return foundAndClickedInstall
```



crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09 10:13:38

iOS弹框处理

背景：iOS的各种app，有各种类型的弹框，目前主要是通过wda的query和获取源码后bs的find找到过滤，再去点击让弹框消失。其中此处分成了2部分：

- 首次启动后，可能会遇到的弹框：`process_popup_iOS_FirstLaunchBeforeMain`
- 其他正常运行期间，可能会遇到的弹框：`process_popup_iOS`

`process_popup_iOS_FirstLaunchBeforeMain`

```
def process_popup_iOS_FirstLaunchBeforeMain(self, soup=None):
    """
    处理iOS中app，在首次登录，进入主页之前，的弹框
    """

    foundAndProcessedPopup = False
    if not soup:
        curPageXml = self.get_page_source()
        soup = CommonUtils.xmlToSoup(curPageXml)

    # if not foundAndProcessedPopup:
    #     foundAndProcessedPopup = self.iOSProcessPopupUpperRightClose(soup)
    # if not foundAndProcessedPopup:
    #     foundAndProcessedPopup = self.iOSProcessPopupUpperRightAlertClose(soup)
    # Note: put common upper right close, for some app, eg 必要, before show main page, exist these popup
    if not foundAndProcessedPopup:
        # foundAndProcessedPopup = self.iOSProcessPopupUpperRightCommonClose(soup)
        foundAndProcessedPopup = self.iOSProcessPopupCommonNameContainClose(soup)

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupPossibleCloseButton(soup)

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupUnderCloseButton(soup)

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupIKnow(soup)

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessCommonUserAgreementAgree(soup)

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessUserPrivacyProtocol(soup)

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessAlertUseWirelessData(soup)

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessCommonAlertAllow(soup)
```

```

if not foundAndProcessedPopup:
    foundAndProcessedPopup = self.iOSProcessCommonAlertOk(soup)

if not foundAndProcessedPopup:
    foundAndProcessedPopup = self.iOSProcessRightUpperJumpOver(soup)

if not foundAndProcessedPopup:
    foundAndProcessedPopup = self.iOSProcessPopupCommonSave(soup)

if not foundAndProcessedPopup:
    foundAndProcessedPopup = self.iOSProcessPopupCommonConfirmButton(soup)

if not foundAndProcessedPopup:
    foundAndProcessedPopup = self.iOSProcessPopupImmediatelyExperience(soup)

return foundAndProcessedPopup

```

process_popup_iOS

```

def process_popup_iOS(self, curPageXml):
    foundAndProcessedPopup = False

    soup = CommonUtils.xmlToSoup(curPageXml)

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.process_popup_iOS_FirstLaunchBeforeMain(soup)

    if self.isSpecialiOSApp_kags:
        if not foundAndProcessedPopup:
            foundAndProcessedPopup = self.iOSProcessKagsPopupStillNotLogin(soup)

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupPhoneCall(soup)

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessSettingsAllowUseLocation(soup)

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupCertificateError(soup)

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupNetworkNotStableRetry(soup)

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupNetworkNotStableRefresh(soup)

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupSystemAbnormalRetryRefresh(soup)

    if not foundAndProcessedPopup:

```

```

        foundAndProcessedPopup = self.iOSProcessPopupServerBusyLaterRetry(soup)

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupWeixinLogin(soup)

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupJumpThirdApp(soup)

    if not foundAndProcessedPopup:
        # foundAndProcessedPopup = self.iOSProcessPopupGetInfoFail(soup)
        foundAndProcessedPopup = self.iOSProcessPopupDoSomeFail(soup)

    if not foundAndProcessedPopup:
        # foundAndProcessedPopup = self.iOSProcessPopupMiniprogramAuthority(soup)
        foundAndProcessedPopup = self.iOSProcessPopupMiniprogramGetYour(soup)

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupWeixinAuthorizeLocation(soup)

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupMiniprogramWarning(soup)

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupMiniprogramDisclaimer(soup)

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessWillOpenNonOfficialPage(soup)

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessAlertNoteConfrim(soup)

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessAlertCancel(soup)

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessAlertCommonGiveup(soup)

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupPhotoCamera(soup)

    # 注: 此逻辑经常误判, 没有弹框误以为有弹框, 所以放弃此逻辑
    # if not foundAndProcessedPopup:
    #     foundAndProcessedPopup = self.iOSProcessCommonPopupWindowUpperRightClose(soup)

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupSovereignHasRead(soup)

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupCommonWindowCancel(soup)

    return foundAndProcessedPopup

```

具体实现:

iOSProcessPopupImmediatelyExperience

```

def iOSProcessPopupImmediatelyExperience(self, soup):
    """
        弹框 other->other-button name=立即体验
    """

    """
        米家 弹框 立即体验:
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
        " x="0" y="0" width="414" height="736">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
            true" x="0" y="0" width="414" height="736">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visib
                le="true" x="0" y="0" width="414" height="736">
                    <XCUIElementTypeImage type="XCUIElementTypeImage" name="sort_guide
                    _icon" enabled="true" visible="false" x="54" y="115" width="306" height="310"/>
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value=
                    "设备排序全新升级" name="设备排序全新升级" label="设备排序全新升级" enabled="true" visible="tru
                    e" x="111" y="426" width="192" height="34"/>
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value=
                    "1.新增快速置顶，排序操作更方便 2.顶部排序固定，不会被随意打乱" name="1.新增快速置顶，排序操作更方
                    便 2.顶部排序固定，不会被随意打乱" label="1.新增快速置顶，排序操作更方便 2.顶部排序固定，不会被随
                    意打乱" enabled="true" visible="true" x="95" y="468" width="224" height="42"/>
                    <XCUIElementTypeButton type="XCUIElementTypeButton" name="立即体验"
                    label="立即体验" enabled="true" visible="true" x="24" y="657" width="366" height="43"/>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    """

    immediatelyExperienceChainList = [
        {
            "tag": "XCUIElementTypeOther",
            "attrs": self.FullScreenAttrDict
        },
        {
            "tag": "XCUIElementTypeOther",
            "attrs": self.FullScreenAttrDict
        },
        {
            "tag": "XCUIElementTypeButton",
            "attrs": {"enabled": "true", "visible": "true", "name": "立即体验"}
        },
    ]
    foundAndProcessedPopup = self.findAndClickCenterPosition(immediatelyExperienceChainList
        , soup)
    return foundAndProcessedPopup

```

iOSProcessPopupCommonConfirmButton

```

def iOSProcessPopupCommonConfirmButton(self, soup):
    """
        other下面other下的button name="确定"
    """
    foundAndProcessedPopup = False
    """
        米家 弹框 米家隐私政策 确定:
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
    " x="0" y="0" width="414" height="736">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
    true" x="41" y="243" width="332" height="250">
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="米家
隐私政策" name="米家隐私政策" label="米家隐私政策" enabled="true" visible="true" x="49" y="26
3" width="316" height="16"/>
                    <XCUIElementTypeTextView type="XCUIElementTypeTextView" value="我们的隐私
政策已于2018年4月28日更新，将会于2018年5月25日生效。我们对隐私政策进行了详细修订，从该日期开始，这一隐私政策能够提供有关我们如何管理你在使用所有米家产品和服务时透露的个人信息的隐私详情，特定的米家产品或服务提供独立的隐私政策除外。&#10;请花一些时间熟悉我们的隐私权惯例，如果你有任何问题，请告诉我们。&#10;&#10; 米家隐私政策" enabled="true" visible="true" x="49" y="283" width="316" height="145"/>
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visib
le="true" x="41" y="442" width="332" height="1"/>
                    <XCUIElementTypeButton type="XCUIElementTypeButton" name="取消" label="取
消" enabled="true" visible="true" x="41" y="443" width="166" height="50"/>
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visib
le="true" x="206" y="443" width="2" height="50"/>
                    <XCUIElementTypeButton type="XCUIElementTypeButton" name="确定" label="确
定" enabled="true" visible="true" x="207" y="443" width="166" height="50"/>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        """
        confirmChainList = [
            {
                "tag": "XCUIElementTypeOther",
                "attrs": self.FullScreenAttrDict
            },
            {
                "tag": "XCUIElementTypeOther",
                "attrs": {"enabled": "true", "visible": "true"}
            },
            {
                "tag": "XCUIElementTypeButton",
                "attrs": {"enabled": "true", "visible": "true", "name": "确定"}
            },
        ]
        confirmSoup = CommonUtils.bsChainFind(soup, confirmChainList)
        if confirmSoup:
            self.clickElementCenterPosition(confirmSoup)
            foundAndProcessedPopup = True
    return foundAndProcessedPopup

```



iOSProcessPopupCommonSave

```

def iOSProcessPopupCommonSave(self, soup):
    """
        通用的弹框 other下的other下的button name=保存
    """
    foundAndProcessedPopup = False
    """
        米家 弹框 地区选择 保存:
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
    " x="0" y="0" width="414" height="736">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
    true" x="0" y="0" width="414" height="736">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visib
    le="true" x="0" y="0" width="414" height="736">

                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" v
    isible="true" x="0" y="64" width="414" height="88">
                        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="tr
    ue" visible="true" x="10" y="64" width="394" height="36">
                            <XCUIElementTypeImage type="XCUIElementTypeImage" enabl
    ed="true" visible="false" x="10" y="64" width="394" height="36"/>
                            <XCUIElementTypeSearchField type="XCUIElementTypeSearchFie
    ld" name="搜索" label="搜索" enabled="true" visible="true" x="18" y="64" width="378" heig
    ht="36"/>
                        </XCUIElementTypeOther>
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="tru
    e" visible="true" x="0" y="110" width="414" height="10"/>
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" va
    lue="选择你的使用国家或地区" name="选择你的使用国家或地区" label="选择你的使用国家或地区" enabl
    ed="true" visible="true" x="22" y="120" width="368" height="32"/>
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="tru
    e" visible="true" x="22" y="151" width="392" height="1"/>
                </XCUIElementTypeOther>
                <XCUIElementTypeTable type="XCUIElementTypeTable" enabled="true" v
    isible="true" x="0" y="152" width="414" height="450">

                    </XCUIElementTypeTable>
                    <XCUIElementTypeButton type="XCUIElementTypeButton" value="1" name
    ="checkbox unchecked" label="checkbox unchecked" enabled="true" visible="true" x="91" y="6
    26" width="20" height="20"/>
                    <XCUIElementTypeButton type="XCUIElementTypeButton" name="我已阅读
    用户协议和隐私政策" label="我已阅读 用户协议和隐私政策" enabled="true" visible="true" x="119"
    y="622" width="176" height="28"/>
                    <XCUIElementTypeButton type="XCUIElementTypeButton" name="保存" lab
    el="保存" enabled="true" visible="true" x="24" y="672" width="366" height="42"/>
                </XCUIElementTypeOther>
    """
    commonSaveChainList = [
        {
            "tag": "XCUIElementTypeOther",
            "attrs": self.FullScreenAttrDict
    ]

```

```

        },
        {
            "tag": "XCUIElementTypeOther",
            "attrs": self.FullScreenAttrDict
        },
        {
            "tag": "XCUIElementTypeButton",
            "attrs": {"enabled": "true", "visible": "true", "name": "保存"}
        }
    ],
commonSaveSoup = CommonUtils.bsChainFind(soup, commonSaveChainList)
if commonSaveSoup:
    self.clickElementCenterPosition(commonSaveSoup)
    foundAndProcessedPopup = True
return foundAndProcessedPopup

```

iOSProcessPopupCommonWindowCancel

```

def iOSProcessPopupCommonWindowCancel(self, soup):
    """
        通用的弹框 window下的other下的button name=取消
    """
    foundAndProcessedPopup = False
    """
        斑马AI课 弹框 该音频为专属音频 取消:
        <XCUIElementTypeWindow type="XCUIElementTypeWindow" enabled="true" visible="true" x="0" y="0" width="414" height="736">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="736">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="0" width="414" height="736"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="20" width="414" height="716"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="30" y="178" width="354" height="380">
                    <XCUIElementTypeImage type="XCUIElementTypeImage" name="ZBEFMArtImage" enabled="true" visible="false" x="93" y="198" width="228" height="184"/>
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="该音频为专属音频, 仅对英语S2系统课用户开放" name="该音频为专属音频, 仅对英语S2系统课用户开放" label="该音频为专属音频, 仅对英语S2系统课用户开放" enabled="true" visible="true" x="57" y="395" width="300" height="43"/>
                    <XCUIElementTypeButton type="XCUIElementTypeButton" name="查看课程详情" label="查看课程详情" enabled="true" visible="true" x="79" y="454" width="256" height="44"/>
                    <XCUIElementTypeButton type="XCUIElementTypeButton" name="取消" label="取消" enabled="true" visible="true" x="191" y="510" width="32" height="30"/>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeWindow>
    """
    commonCancelChainList = [

```

```

    {
        "tag": "XCUIElementTypeWindow",
        "attrs": self.FullScreenAttrDict
    },
    {
        "tag": "XCUIElementTypeOther",
        # "attrs": self.FullScreenAttrDict
        "attrs": {"enabled":"true", "visible":"true"}
    },
    {
        "tag": "XCUIElementTypeButton",
        "attrs": {"enabled":"true", "visible":"true", "name":"取消"}
    },
]
commonCancelSoup = CommonUtils bsChainFind(soup, commonCancelChainList)
if commonCancelSoup:
    self.clickElementCenterPosition(commonCancelSoup)
    foundAndProcessedPopup = True
return foundAndProcessedPopup

```

iOSProcessPopupSovereignHasRead

```

def iOSProcessPopupSovereignHasRead(self, soup):
"""
    必要 弹框 耷已阅
"""
foundAndProcessedPopup = False
"""
    必要 弹框 耷已阅:
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
        " x="0" y="0" width="414" height="736">
            <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true"
            true" x="44" y="156" width="326" height="424">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="抢福
利活动说明" name="抢福利活动说明" label="抢福利活动说明" enabled="true" visible="true" x="149"
            y="165" width="117" height="20"/>
                <XCUIElementTypeTextView type="XCUIElementTypeTextView" value="抢福利活
动全新上线啦！！！&#10;当您获得特权金、参团卡、全民拼卡、返现卡、金币等资产后，请您尽快使用，很可能
会被好友抢走哦！&#10;本期重磅推出&lt;保护罩&gt;道具卡！您的资产可受到保护不被抢走，快去使用金币在每
日签到页兑换吧！！&#10;新玩法需要用户更新APP，如果您更新APP后，仍然无法参与活动，有可能APP正在发版
中，请您等待。" enabled="true" visible="true" x="66" y="204" width="282" height="292"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="朕已阅" label
                ="朕已阅" enabled="true" visible="true" x="82" y="513" width="250" height="50"/>
            </XCUIElementTypeImage>
        </XCUIElementTypeOther>
"""
hasReadChainList = [
{
    "tag": "XCUIElementTypeOther",
    "attrs": {"enabled":"true", "visible":"true", "x":"0", "y":"0", "width": "%s" %
self.X, "height": "%s" % self.totalY}

```

```

        },
        {
            "tag": "XCUIElementTypeImage",
            "attrs": {"enabled": "true", "visible": "true"}
        },
        {
            "tag": "XCUIElementTypeButton",
            "attrs": {"enabled": "true", "visible": "true", "name": "朕已阅"}
        }
    ],
    hasReadCloseSoup = CommonUtils.bsChainFind(soup, hasReadChainList)
    if hasReadCloseSoup:
        self.clickElementCenterPosition(hasReadCloseSoup)
        foundAndProcessedPopup = True
    return foundAndProcessedPopup

```

iOSProcessAlertCommonGiveup

```

def iOSProcessAlertCommonGiveup(self, soup):
    """
        通用逻辑：对于alert弹框， 找里面的 放弃 button，并点击
    """
    foundAndProcessedPopup = False
    """
        京东金融 弹框 是否放弃注册？
        <XCUIElementTypeAlert type="XCUIElementTypeAlert" name="是否放弃注册?" label="是否放弃注册?" enabled="true" visible="true" x="72" y="316" width="270" height="105">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="316" width="270" height="105">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="316" width="270" height="105">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="316" width="270" height="105">
                        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="316" width="270" height="105"/>
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="316" width="270" height="105">
                        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="316" width="270" height="105"/>
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="316" width="270" height="105"/>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="316" width="270" height="105">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="316" width="270" height="60">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="316" width="270" height="60"/>
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="是否放弃注册?" name="是否放弃注册?" label="是否放弃注册?" enabled="true" visible="true" x="72" y="316" width="270" height="60"/>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeAlert>
    
```

```

        le="true" x="88" y="335" width="238" height="21"/>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true"
e" visible="true" x="72" y="376" width="270" height="1">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled=
"true" visible="false" x="72" y="376" width="270" height="1"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled=
"true" visible="true" x="72" y="376" width="270" height="1"/>
            </XCUIElementTypeOther>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true"
e" visible="true" x="72" y="376" width="270" height="45">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled=
"true" visible="true" x="72" y="376" width="270" height="45">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enab
led="true" visible="true" x="72" y="376" width="270" height="45">
                    <XCUIElementTypeOther type="XCUIElementTypeOther"
enabled="true" visible="true" x="72" y="376" width="135" height="45">
                        <XCUIElementTypeButton type="XCUIElementTypeBu
tton" name="放弃" label="放弃" enabled="true" visible="true" x="72" y="376" width="135" hei
ght="45"/>
                    </XCUIElementTypeOther>
                    <XCUIElementTypeOther type="XCUIElementTypeOther"
enabled="true" visible="true" x="207" y="376" width="1" height="45">
                        <XCUIElementTypeOther type="XCUIElementTypeOth
er" enabled="true" visible="false" x="207" y="376" width="1" height="45"/>
                        <XCUIElementTypeOther type="XCUIElementTypeOth
er" enabled="true" visible="true" x="207" y="376" width="1" height="45"/>
                        </XCUIElementTypeOther>
                        <XCUIElementTypeOther type="XCUIElementTypeOther"
enabled="true" visible="true" x="207" y="376" width="135" height="45">
                            <XCUIElementTypeButton type="XCUIElementTypeBu
tton" name="继续注册" label="继续注册" enabled="true" visible="true" x="207" y="376" width="135"
height="45"/>
                        </XCUIElementTypeOther>
                    </XCUIElementTypeOther>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    </XCUIElementTypeAlert>
    """
giveupChainList = [
    {
        "tag": "XCUIElementTypeAlert",
        "attrs": {"enabled": "true", "visible": "true"}
    },
    {
        "tag": "XCUIElementTypeOther",
        "attrs": {"enabled": "true", "visible": "true"}
    },
    {

```

```

        "tag": "XCUIElementTypeButton",
        "attrs": {"enabled":"true", "visible":"true", "name":"放弃"}
    },
]
alertGiveupSoup = CommonUtils.bsChainFind(soup, giveupChainList)
if alertGiveupSoup:
    self.clickElementCenterPosition(alertGiveupSoup)
    foundAndProcessedPopup = True
return foundAndProcessedPopup

```

iOSProcessPopupServerBusyLaterRetry

```

def iOSProcessPopupServerBusyLaterRetry(self, soup):
    foundAndProcessedPopup = False
    """
        京东金融 异常页面 服务器繁忙, 请稍后重试
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
        " x="0" y="0" width="414" height="736">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
            " x="0" y="0" width="414" height="64">
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="com icon bac
                kup u" label="com icon backup u" enabled="true" visible="true" x="0" y="20" width="44" hei
                ght="44"/>
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" enabled="t
                    rue" visible="false" x="80" y="64" width="254" height="24"/>
                </XCUIElementTypeOther>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
                true" x="0" y="64" width="414" height="672">
                    <XCUIElementTypeImage type="XCUIElementTypeImage" name="network_err_do
                    g" enabled="true" visible="false" x="152" y="164" width="110" height="110"/>
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="服务
                    器繁忙, 请稍后重试" name="服务器繁忙, 请稍后重试" label="服务器繁忙, 请稍后重试" en
                    abled="true" visible="true" x="0" y="304" width="414" height="17"/>
                </XCUIElementTypeOther>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
                false" x="0" y="686" width="414" height="50">
                    <XCUIElementTypeCollectionView type="XCUIElementTypeCollectionView" en
                    abled="true" visible="false" x="0" y="686" width="414" height="50"/>
                    <XCUIElementTypeButton type="XCUIElementTypeButton" enabled="true" vis
                    ible="false" x="0" y="686" width="414" height="50"/>
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" enabled="t
                    rue" visible="false" x="16" y="701" width="382" height="20"/>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        """
    busyRetryChainList = [
        {
            "tag": "XCUIElementTypeOther",
            "attrs": {"enabled":"true", "visible":"true", "x":0, "y":0, "width": "%s" %
            self.X, "height": "%s" % self.totalY}
        },
    ]

```

```

    {
        "tag": "XCUIElementTypeOther",
        "attrs": {"enabled": "true", "visible": "true", "x": "0", "width": "%s" % self.X}
    },
    {
        "tag": "XCUIElementTypeStaticText",
        "attrs": {"enabled": "true", "visible": "true", "value": "服务器繁忙，请稍后重试"}
    },
]
busyRetrySoup = CommonUtils.bsChainFind(soup, busyRetryChainList)
if busyRetrySoup:
    # find back button
    parentOtherSoup = busyRetrySoup.parent
    parentParentOtherSoup = parentOtherSoup.parent
    if parentParentOtherSoup:
        backupP = re.compile("backup") # "com icon backup u"
        backupSoup = parentParentOtherSoup.find(
            "XCUIElementTypeButton",
            attrs={"visible": "true", "enabled": "true", "name": backupP}
        )
        if backupSoup:
            self.clickElementCenterPosition(backupSoup)
            foundAndProcessedPopup = True
return foundAndProcessedPopup

```

iOSProcessPopupSystemAbnormalRetryRefresh

```

def iOSProcessPopupSystemAbnormalRetryRefresh(self, soup):
    foundAndProcessedPopup = False
    """
        京东金融 异常页面 系统正在开小差, 请稍后再试 再刷新下:
        <XCUIElementTypeOther type="XCUIElementTypeOther" name="系统正在开小差, 请稍后再试 再刷新下" label="系统正在开小差, 请稍后再试 再刷新下" enabled="true" visible="true" x="0" y="64" width="414" height="672">
            <XCUIElementTypeOther type="XCUIElementTypeOther" name="系统正在开小差, 请稍后再试 再刷新下" label="系统正在开小差, 请稍后再试 再刷新下" enabled="true" visible="true" x="0" y="64" width="414" height="672">
                <XCUIElementTypeOther type="XCUIElementTypeOther" name="系统正在开小差, 请稍后再试" label="系统正在开小差, 请稍后再试" enabled="true" visible="true" x="114" y="240" width="186" height="136">
                    <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="152" y="240" width="110" height="111"/>
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="系统正在开小差, 请稍后再试" name="系统正在开小差, 请稍后再试" label="系统正在开小差, 请稍后再试" enabled="true" visible="true" x="114" y="360" width="186" height="16"/>
                </XCUIElementTypeOther>
                <XCUIElementTypeOther type="XCUIElementTypeOther" name="再刷新下" label="再刷新下" enabled="true" visible="true" x="147" y="375" width="120" height="61">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" name="再刷新下" label="再刷新下" enabled="true" visible="true" x="147" y="395" width="120" height="41"/>
                </XCUIElementTypeOther>

```

```

        </XCUIElementTypeOther>
    </XCUIElementTypeOther>
"""

retryRefreshChainList = [
    {
        "tag": "XCUIElementTypeOther",
        "attrs": {"enabled": "true", "visible": "true", "x": "0", "width": "%s" % self.X}
    },
    {
        "tag": "XCUIElementTypeOther",
        "attrs": {"enabled": "true", "visible": "true", "x": "0", "width": "%s" % self.X}
    },
    {
        "tag": "XCUIElementTypeOther",
        "attrs": {"enabled": "true", "visible": "true", "name": "再刷新下"}
    },
]
retryRefreshSoup = CommonUtils.bsChainFind(soup, retryRefreshChainList)
if retryRefreshSoup:
    self.clickElementCenterPosition(retryRefreshSoup)
    foundAndProcessedPopup = True
return foundAndProcessedPopup

```

iOSProcessPopupUnderCloseButton

```

def iOSProcessPopupUnderCloseButton(self, soup):
"""
    弹框 关闭按钮在弹框下面的 尤其是 name=""
"""

foundAndProcessedPopup = False
"""

    途虎养车 弹框 关闭按钮在下面 新人618全品类消费券:
        <XCUIElementTypeWindow type="XCUIElementTypeWindow" enabled="true" visible="true" x="0" y="0" width="414" height="736">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="736">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="736">
                    <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="false" x="72" y="190" width="270" height="356"/>
                    <XCUIElementTypeButton type="XCUIElementTypeButton" enabled="true" visible="true" x="72" y="190" width="270" height="356"/>
                    <XCUIElementTypeButton type="XCUIElementTypeButton" name="" label="" enabled="true" visible="true" x="187" y="589" width="40" height="41"/>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    </XCUIElementTypeWindow>

```

```

"""
specialChar = ""
underCloseChainList = [
    {
        "tag": "XCUIElementTypeWindow",
        "attrs": self.FullScreenAttrDict,
    },
    {
        "tag": "XCUIElementTypeOther",
        "attrs": self.FullScreenAttrDict,
    },
    {
        "tag": "XCUIElementTypeButton",
        "attrs": {"enabled": "true", "visible": "true", "name": specialChar}
    },
]
underCloseSoup = CommonUtils.bsChainFind(soup, underCloseChainList)
if underCloseSoup:
    self.clickElementCenterPosition(underCloseSoup)
    foundAndProcessedPopup = True

return foundAndProcessedPopup

```

iOSProcessPopupPossibleCloseButton

```

def isPopupWindowSize(self, curSize):
    """判断一个soup的宽高大小是否是弹框类窗口(Image, Other等) 的大小"""
    # global FullScreenSize
    FullScreenSize = self.X * self.totalY
    curSizeRatio = curSize / FullScreenSize # 0.289
    PopupWindowSizeMinRatio = 0.25
    # PopupWindowSizeMaxRatio = 0.9
    PopupWindowSizeMaxRatio = 0.8
    # isSizeValid = curSizeRatio >= MinPopupWindowSizeRatio
    # is popup like window, size should large enough, but should not full screen
    isSizeValid = PopupWindowSizeMinRatio <= curSizeRatio <= PopupWindowSizeMaxRatio
    return isSizeValid

def isNormalButtonSize(self, curSize):
    """判断一个soup的宽高大小是否是普通的按钮大小"""
    NormalButtonSizeMin = 30*30
    NormalButtonSizeMax = 100*100
    isNormalSize = NormalButtonSizeMin <= curSize <= NormalButtonSizeMax
    return isNormalSize

def iOSProcessPopupPossibleCloseButton(self, soup):
    """
    必要 弹框 喜单限时福利 右上角 关闭按钮
    特殊: 但是内部无有效识别内容, 比如name中包含close这类条件
    暂时只能特殊但通用处理:
    """

```

```
window -> other -> button
且 button 的 w 和 h 基本一致，后者差距小于 w 和 h 最大值的 10%
以及 w 和 h 都在一定（普通关闭按钮的大小）范围，比如 20-60
```

必要 我的 蒙层弹框 首次使用引导页：

基于上面逻辑：

```
window->other->button
```

button 条件不变

基础上，底层判断逻辑是：

next 的 sibling 后面，有且只有一个 button，且有 name

且按钮宽高大小在合理范围，比如普通按钮的大小，算作为 $30 \times 30 < \text{按钮面积} < 100 \times 100$

意思是：引导页往往伴随着一个普通大小的按钮，供用户点击

必要 弹框 推荐开通以下授权 关闭按钮在弹框下面：

和前面逻辑略有不同

```
Application -> window -> Other
```

button 条件不变

其他条件类似于第一个，但是是 prev 的 sibling 是个 Other 元素，且面积是弹框类大小

"""

```
foundAndProcessedPopup = False
```

"""

必要 弹框 首单限时福利 右上角 关闭按钮：

```
<XCUIElementTypeWindow type="XCUIElementTypeWindow" enabled="true" visible="true" x="0" y="0" width="414" height="736">
    ...
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="736">
            <XCUIElementTypeButton type="XCUIElementTypeButton" enabled="true" visible="true" x="339" y="122" width="31" height="32"/>
            <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="44" y="174" width="326" height="388"/>
        </XCUIElementTypeOther>
    </XCUIElementTypeWindow>
```

必要 我的 蒙层弹框 首次使用引导页：

```
<XCUIElementTypeWindow type="XCUIElementTypeWindow" enabled="true" visible="true" x="0" y="0" width="414" height="736">
    ...
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="736">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="208"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="392" width="414" height="344"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="101" y="406" width="296" height="186">
                <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="false" x="101" y="406" width="296" height="186"/>
                <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="false" x="236" y="441" width="25" height="25"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton" enabled="true" visible="true" x="367" y="459" width="25" height="25"/>
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="
```

```

"发现新功能" name="发现新功能" label="发现新功能" enabled="true" visible="true" x="114" y="46
8" width="88" height="18"/>
    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value=
    "必要地震预警通知功能（测试版）上线！ 曾提前61秒成功为成都预警四川长宁地震的成都高新减灾研究所与必
    要战略合作，为您的安全保驾护航！" name="必要地震预警通知功能（测试版）上线！ 曾提前61秒成功为成都
    预警四川长宁地震的成都高新减灾研究所与必要战略合作，为您的安全保驾护航！" label="必要地震预警通知功
    能（测试版）上线！ 曾提前61秒成功为成都预警四川长宁地震的成都高新减灾研究所与必要战略合作，为您的安
    全保驾护航！" enabled="true" visible="true" x="114" y="488" width="278" height="51"/>
        <XCUIElementTypeButton type="XCUIElementTypeButton" name="开启预警"
        label="开启预警" enabled="true" visible="true" x="316" y="547" width="67" height="32"/>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>

    必要 弹框 推荐开通以下授权 关闭按钮在弹框下面：：
    <XCUIElementTypeApplication type="XCUIElementTypeApplication" name="必要" label
    ="必要" enabled="true" visible="true" x="0" y="0" width="414" height="736">
        <XCUIElementTypeWindow type="XCUIElementTypeWindow" enabled="true" visible
        ="true" x="0" y="0" width="414" height="736">
            . . .
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visib
            le="true" x="44" y="233" width="326" height="270">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value=
                "推荐开通以下授权" name="推荐开通以下授权" label="推荐开通以下授权" enabled="true" visible="tru
                e" x="53" y="257" width="308" height="24"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" v
                isible="true" x="86" y="324" width="242" height="58">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="tru
                    e" visible="true" x="85" y="324" width="244" height="58">
                        <XCUIElementTypeImage type="XCUIElementTypeImage" enabled=
                        "true" visible="false" x="85" y="331" width="45" height="45"/>
                        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText"
                        " value="定位" name="定位" label="定位" enabled="true" visible="true" x="140" y="324" width=
                        "33" height="19"/>
                        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText"
                        " value="开启定位通知将在推荐、配送等方面为您提供更精准服务。" name="开启定位通知将在推荐、配送等
                        方面为您提供更精准服务。" label="开启定位通知将在推荐、配送等方面为您提供更精准服务。" enabled="t
                        rue" visible="true" x="140" y="353" width="189" height="29"/>
                    </XCUIElementTypeOther>
                </XCUIElementTypeOther>
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="开启"
                label="开启" enabled="true" visible="true" x="69" y="429" width="276" height="50"/>
            </XCUIElementTypeOther>
            <XCUIElementTypeButton type="XCUIElementTypeButton" enabled="true" vis
            ible="true" x="191" y="538" width="32" height="31"/>
        . . .
        # noNameP = re.compile("??")
        noNameP = False
        possibleCloseChainList = [
            {
                "tag": "XCUIElementTypeWindow",
                "attrs": {"enabled": "true", "visible": "true", "x": "0", "y": "0", "width": "%s" %
                self.X, "height": "%s" % self.totalY}
            },
        ]
    
```

```

    {
        "tag": "XCUIElementTypeOther",
        "attrs": {"enabled": "true", "visible": "true", "x": "0", "y": "0", "width": "%s" %
self.X, "height": "%s" % self.totalY}
    },
    {
        "tag": "XCUIElementTypeButton",
        "attrs": {"enabled": "true", "visible": "true", "name": noNameP}
    },
],
possibleCloseSoup = CommonUtils.bsChainFind(soup, possibleCloseChainList)

isCloseUnderPopup = False
if not possibleCloseSoup:
    # 尝试找 是否是 关闭按钮在弹框下面的 弹框
    closeUnderPopupChainList = [
        {
            "tag": "XCUIElementTypeApplication",
            "attrs": {"enabled": "true", "visible": "true", "x": "0", "y": "0", "width": "%
s" % self.X, "height": "%s" % self.totalY}
        },
        {
            "tag": "XCUIElementTypeWindow",
            "attrs": {"enabled": "true", "visible": "true", "x": "0", "y": "0", "width": "%
s" % self.X, "height": "%s" % self.totalY}
        },
        {
            "tag": "XCUIElementTypeButton",
            "attrs": {"enabled": "true", "visible": "true", "name": noNameP}
        },
    ]
possibleCloseSoup = CommonUtils.bsChainFind(soup, closeUnderPopupChainList)
if possibleCloseSoup:
    isCloseUnderPopup = True

if possibleCloseSoup:
    # check is match close button rule or not
    isValidSize = False
    isAlmostSameWH = False
    # isNextSiblingImage = False
    # isImageSizeValid = False
    isPossibleClose = True

    if isPossibleClose:
        soupAttrDict = possibleCloseSoup.attrs # {'enabled': 'true', 'height': '32', '
type': 'XCUIElementTypeButton', 'visible': 'true', 'width': '31', 'x': '339', 'y': '122'}
        logging.debug("possibleCloseSoup soupAttrDict=%s", soupAttrDict)
        # x = int(soupAttrDict["x"])
        # y = int(soupAttrDict["y"])
        width = int(soupAttrDict["width"])
        height = int(soupAttrDict["height"])
        # ButtonMinWH = 20
        # ButtonMinWH = 30

```

```

        ButtonMinWH = 25
        ButtonMaxWH = 60
        isValidWidth = ButtonMinWH <= width <= ButtonMaxWH
        isValidHeight = ButtonMinWH <= height <= ButtonMaxWH
        isValidSize = isValidWidth and isValidHeight

        isPossibleClose = isValidSize

    if isPossibleClose:
        isAlmostSameWH = False
        isSameWH = width == height
        if isSameWH:
            isAlmostSameWH = isSameWH
        else:
            maxWH = max(width, height)
            diffWH = abs(width - height)
            MaxAllowDiffRatio = 0.1
            curDiffRatio = diffWH / maxWH
            isValidDiff = curDiffRatio <= MaxAllowDiffRatio
            isAlmostSameWH = isValidDiff

        isPossibleClose = isAlmostSameWH

    if isPossibleClose:
        # check next sibling is large Image
        # nextSiblingeSoup = possibleCloseSoup.next_sibling # '\n'
        # nextSiblingeSoupList = possibleCloseSoup.next_siblings
        nextSiblingeSoupGenerator = possibleCloseSoup.next_siblings
        nextSiblingeSoupList = list(nextSiblingeSoupGenerator)

        hasLargeImage = CommonUtils.isContainSpecificSoup(nextSiblingeSoupList, "XCUIElementTypeImage", self.isPopupWindowSize)
        isPossibleClose = hasLargeImage

    if not isPossibleClose:
        hasNormalButton = CommonUtils.isContainSpecificSoup(nextSiblingeSoupList, "XCUIElementTypeButton", self.isNormalButtonSize)
        isPossibleClose = hasNormalButton

    if not isPossibleClose:
        if isCloseUnderPopup:
            # prevSiblingSoupGenerator = possibleCloseSoup.previous_siblings
            # prevPrevSiblingSoupList = list(prevSiblingSoupGenerator)
            # hasLargeOther = CommonUtils.isContainSpecificSoup(prevPrevSiblingSoupList, "XCUIElementTypeOther", self.isPopupWindowSize)
            prevSiblingSoup = possibleCloseSoup.previous_sibling # '\n'
            if prevSiblingSoup:
                prevPrevSiblingSoup = prevSiblingSoup.previous_sibling # XCUIElementTypeOther
                prevPrevSiblingSoupList = [prevPrevSiblingSoup]
                hasLargeOther = CommonUtils.isContainSpecificSoup(prevPrevSiblingSoupList, "XCUIElementTypeOther", self.isPopupWindowSize)
                isPossibleClose = hasLargeOther

```

```

    if isPossibleClose:
        foundAndProcessedPopup = self.findAndClickButtonElementBySoup(possibleCloseSoup)
    )

    return foundAndProcessedPopup

```

iOSProcessPopupCommonNameContainClose

```

# def iOSProcessPopupUpperRightCommonClose(self, soup):
def iOSProcessPopupCommonNameContainClose(self, soup):
    foundAndProcessedPopup = False
    """
        京东金融 登录页 弹框 右上角 关闭按钮:
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
    " x="0" y="0" width="414" height="736">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
true" x="0" y="0" width="414" height="64">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" enabled="t
rue" visible="false" x="65" y="20" width="284" height="44"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="com icon bla
ck close u" label="com icon black close u" enabled="true" visible="true" x="370" y="20" wi
dth="44" height="44"/>
            </XCUIElementTypeOther>

        京东金融 tab2财富 弹框 我是Max alertClose:
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
    " x="0" y="0" width="414" height="736">
            . . .
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
true" x="0" y="0" width="414" height="736">
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="jr fianne
ncing alertClose@3x" label="jr fianneing alertClose@3x" enabled="true" visible="true" x="327" y
="132" width="26" height="25"/>

        必要 弹框 右上角关闭按钮 name中有close:
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
    " x="0" y="0" width="414" height="736">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
false" x="0" y="0" width="414" height="736"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
true" x="44" y="150" width="327" height="436">
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="rights cl
ose white" label="rights close white" enabled="true" visible="true" x="338" y="150" width="32
" height="32"/>
                <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visib
le="true" x="44" y="199" width="326" height="387">
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value=
"01 : 58 : 31 . 6 后过期" name="01 : 58 : 31 . 6 后过期" label="01 : 58 : 31 . 6 后过期" ena
bled="true" visible="true" x="137" y="476" width="140" height="18"/>

```

```

                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="137" y="495" width="140" height="2"/>
            </XCUIElementTypeImage>
        </XCUIElementTypeOther>
    </XCUIElementTypeOther>

    必要 验证码登录页 左上角 关闭 按钮 name中含close:
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="736">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="64">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="64"/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="login close" label="login close" enabled="true" visible="true" x="10" y="20" width="44" height="44"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="293" y="20" width="111" height="44">
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="密码登录" label="密码登录" enabled="true" visible="true" x="293" y="20" width="111" height="44"/>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    <!--
    commonCloseP = re.compile("close", flags=re.I)
    # com icon black close u
    # jr fianneing alertClose@3x
    # rights close white
    # login close
    commonCloseChainList = [
        {
            "tag": "XCUIElementTypeOther",
            "attrs": {"enabled": "true", "visible": "true", "x": "0", "y": "0", "width": "%s" % self.X, "height": "%s" % self.totalY}
        },
        {
            "tag": "XCUIElementTypeOther",
            "attrs": {"enabled": "true", "visible": "true", "x": "0", "y": "0", "width": "%s" % self.X, "height": "%s" % self.totalY}
            # "attrs": {"enabled": "true", "visible": "true", "x": "0", "y": "0", "width": "%s" % self.X}
            # "attrs": {"enabled": "true", "visible": "true", "x": "0", "y": "0", "width": "%s" % self.X}
            "attrs": {"enabled": "true", "visible": "true"}
        },
        {
            "tag": "XCUIElementTypeButton",
            "attrs": {"enabled": "true", "visible": "true", "name": commonCloseP}
        },
    ]
    commonCloseSoup = CommonUtils.bsChainFind(soup, commonCloseChainList)
    if commonCloseSoup:
        # self.clickElementCenterPosition(commonCloseSoup) # sometime not work
        # foundAndProcessedPopup = True

        # so change to wda query element then click
    foundAndProcessedPopup = self.findAndClickButtonElementBySoup(commonCloseSoup)
-->
```

```
return foundAndProcessedPopup
```

iOSProcessPopupUpperRightAlertClose

```
# def iOSProcessPopupUpperRightAlertClose(self, soup):
#     foundAndProcessedPopup = False
#
#     """
#         京东金融 tab2财富 弹框 我是Max alertClose:
#             <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="736">
#                 ...
#                 <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="736">
#                     <XCUIElementTypeButton type="XCUIElementTypeButton" name="jr fianncing alertClose@3x" label="jr fianncing alertClose@3x" enabled="true" visible="true" x="327" y="132" width="26" height="25"/>
#                 """
#         alertCloseP = re.compile("alertClose")
#         alertCloseChainList = [
#             {
#                 "tag": "XCUIElementTypeOther",
#                 "attrs": {"enabled": "true", "visible": "true", "x": "0", "y": "0", "width": "%s" % self.X, "height": "%s" % self.totalY}
#             },
#             {
#                 "tag": "XCUIElementTypeOther",
#                 "attrs": {"enabled": "true", "visible": "true", "x": "0", "y": "0", "width": "%s" % self.X, "height": "%s" % self.totalY}
#             },
#             {
#                 "tag": "XCUIElementTypeButton",
#                 "attrs": {"enabled": "true", "visible": "true", "name": alertCloseP}
#             },
#         ]
#         alertCloseSoup = CommonUtils.bsChainFind(soup, alertCloseChainList)
#         if alertCloseSoup:
#             self.clickElementCenterPosition(alertCloseSoup)
#             foundAndProcessedPopup = True
#     return foundAndProcessedPopup

# def iOSProcessPopupUpperRightClose(self, soup):
#     foundAndProcessedPopup = False
#
#     """
#         京东金融 登录页 弹框 右上角 关闭按钮:
#             <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="736">
#                 <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="64">
#                     <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" enabled="true" visible="false" x="65" y="20" width="284" height="44"/>
#                     <XCUIElementTypeButton type="XCUIElementTypeButton" name="com ic
```

```

on black close u" label="com icon black close u" enabled="true" visible="true" x="370" y=
20" width="44" height="44"/>
#
</XCUIElementTypeOther>
#
"""
# containCloseP = re.compile("close") # com icon black close u
# closeChainList = [
#     {
#         "tag": "XCUIElementTypeOther",
#         "attrs": {"enabled":"true", "visible":"true", "x":"0", "y":"0", "width":"%" % self.X, "height":"%" % self.totalY}
#     },
#     {
#         "tag": "XCUIElementTypeOther",
#         "attrs": {"enabled":"true", "visible":"true", "x":"0", "y":"0", "width":"%" % self.X}
#     },
#     {
#         "tag": "XCUIElementTypeButton",
#         "attrs": {"enabled":"true", "visible":"true", "name": containCloseP}
#     },
# ]
closeSoup = CommonUtils.bsChainFind(soup, closeChainList)
if closeSoup:
    self.clickElementCenterPosition(closeSoup)
    foundAndProcessedPopup = True
return foundAndProcessedPopup

```

iOSProcessRightUpperJumpOver

```

def iOSProcessRightUpperJumpOver(self, soup):
    foundAndProcessedPopup = False
    """
        京东金融 登录页 右上角 跳过:
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
" x="0" y="0" width="414" height="736">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
true" x="0" y="0" width="414" height="736">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visib
le="true" x="0" y="0" width="414" height="64">
                        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" enable
d="true" visible="false" x="65" y="20" width="284" height="44"/>
                        <XCUIElementTypeButton type="XCUIElementTypeButton" name="跳过" lab
el="跳过" enabled="true" visible="true" x="344" y="29" width="55" height="26"/>
                    </XCUIElementTypeOther>
                """
    jumpOverChainList = [
        {
            "tag": "XCUIElementTypeOther",
            "attrs": {"enabled":"true", "visible":"true", "x":"0", "y":"0", "width":"%" % self.X, "height":"%" % self.totalY}
        },
    ]

```

```

    {
        "tag": "XCUIElementTypeOther",
        "attrs": {"enabled":"true", "visible":"true", "x":"0", "y":"0", "width":"%s" % self.x, "height":"%s" % self.totalY}
    },
    {
        "tag": "XCUIElementTypeButton",
        "attrs": {"enabled":"true", "visible":"true", "name":"跳过"}
    }
]
jumpOverSoup = CommonUtils.bsChainFind(soup, jumpOverChainList)
if jumpOverSoup:
    self.clickElementCenterPosition(jumpOverSoup)
    foundAndProcessedPopup = True
return foundAndProcessedPopup

```

iOSProcessAlertUseWirelessData

```

def iOSProcessAlertUseWirelessData(self, soup):
    foundAndProcessedPopup = False
    """
    恒易贷 允许使用无线数据 无线局域网与蜂窝移动网络:
        <XCUIElementTypeAlert type="XCUIElementTypeAlert" name="允许“恒易贷”使用无线数据?
    ? " label="允许“恒易贷”使用无线数据? " enabled="true" visible="true" x="72" y="253" width="27
    0" height="230">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible=
    "true" x="72" y="253" width="270" height="230">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visib
    le="true" x="72" y="253" width="270" height="230">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true"
    visible="false" x="72" y="253" width="270" height="230">
                        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="tr
    ue" visible="false" x="72" y="253" width="270" height="230"/>
                        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled=
    "true" visible="false" x="72" y="253" width="270" height="230">
                            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled=
    "true" visible="false" x="72" y="253" width="270" height="230"/>
                            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled=
    "true" visible="false" x="72" y="253" width="270" height="230"/>
                        </XCUIElementTypeOther>
                    </XCUIElementTypeOther>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" v
    isible="true" x="72" y="253" width="270" height="230">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="tr
    ue" visible="true" x="72" y="253" width="270" height="97">
                        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled=
    "true" visible="true" x="72" y="253" width="270" height="97">
                            <XCUIElementTypeStaticText type="XCUIElementTypeStatic
    Text" value="允许“恒易贷”使用无线数据? " name="允许“恒易贷”使用无线数据? " label="允许“恒易贷”使
    用无线数据? " enabled="true" visible="true" x="88" y="273" width="238" height="21"/>
                            <XCUIElementTypeStaticText type="XCUIElementTypeStatic
    Text" value="允许“恒易贷”使用无线数据? " name="允许“恒易贷”使用无线数据? " label="允许“恒易贷”使
    用无线数据? " enabled="true" visible="true" x="88" y="273" width="238" height="21"/>
                        </XCUIElementTypeOther>
                    </XCUIElementTypeOther>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeAlert>
    """

```

```

    Text" value="关闭无线数据时，部分功能可能无法使用。" name="关闭无线数据时，部分功能可能无法使用。
    " label="关闭无线数据时，部分功能可能无法使用。" enabled="true" visible="true" x="88" y="297"
    width="238" height="32"/>
        </XCUIElementTypeOther>
        </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="349" width="270" height="1">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="349" width="270" height="1"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="349" width="270" height="1"/>
        </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="350" width="270" height="133">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="350" width="270" height="133">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="350" width="270" height="133">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="350" width="270" height="44">
                        <XCUIElementTypeButton type="XCUIElementTypeButton" name="无线局域网与蜂窝移动网络" label="无线局域网与蜂窝移动网络" enabled="true" visible="true" x="72" y="350" width="270" height="44"/>
                    </XCUIElementTypeOther>
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="394" width="270" height="1">
                        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="394" width="270" height="1"/>
                        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="394" width="270" height="1"/>
                    </XCUIElementTypeOther>
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="394" width="270" height="45">
                        <XCUIElementTypeButton type="XCUIElementTypeButton" name="仅限无线局域网" label="仅限无线局域网" enabled="true" visible="true" x="72" y="394" width="270" height="45"/>
                    </XCUIElementTypeOther>
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="438" width="270" height="1">
                        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="438" width="270" height="1"/>
                        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="438" width="270" height="1"/>
                    </XCUIElementTypeOther>
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="438" width="270" height="45">
                        <XCUIElementTypeButton type="XCUIElementTypeButton" name="不允许" label="不允许" enabled="true" visible="true" x="72" y="438" width="270" height="45"/>
                    </XCUIElementTypeOther>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    </XCUIElementTypeOther>
</XCUIElementTypeOther>
</XCUIElementTypeOther>
</XCUIElementTypeOther>

```

```

                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    </XCUIElementTypeAlert>
"""

wifiCellularChainList = [
    {
        "tag": "XCUIElementTypeAlert",
        "attrs": {"enabled": "true", "visible": "true"}
    },
    {
        "tag": "XCUIElementTypeOther",
        "attrs": {"enabled": "true", "visible": "true"}
    },
    {
        "tag": "XCUIElementTypeButton",
        "attrs": {"enabled": "true", "visible": "true", "name": "无线局域网与蜂窝移动网络"}
    },
]
wifiCellularSoup = CommonUtils.bsChainFind(soup, wifiCellularChainList)
if wifiCellularSoup:
    # found 无线局域网与蜂窝移动网络 but
    # self.clickElementCenterPosition(wifiCellularSoup) # actually clicked 不允许 !!!
    # foundAndProcessedPopup = True

    # # change to wda query element then click by element
    # curName = wifiCellularSoup.attrs["name"] # 好
    # wifiCellularButtonQuery = {"type": "XCUIElementTypeButton", "enabled": "true", "name": curName}
    # foundAndClicked = self.findAndClickElement(wifiCellularButtonQuery, isShowErrLog=False)
    # foundAndProcessedPopup = foundAndClicked
    foundAndProcessedPopup = self.findAndClickButtonElementBySoup(wifiCellularSoup)
return foundAndProcessedPopup

```

iOSProcessCommonAlertOk

```

def iOSProcessCommonAlertOk(self, soup):
"""
    对于 alert弹框，点击 好
"""

foundAndProcessedPopup = False
"""

    恒易贷 弹框 想访问您的通讯录 好：
        <XCUIElementTypeAlert type="XCUIElementTypeAlert" name="" "恒易贷"想访问您的通讯录
    " label="" "恒易贷"想访问您的通讯录" enabled="true" visible="true" x="72" y="298" width="270"
    height="141">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
            true" x="72" y="298" width="270" height="141">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visib

```

```

        le="true" x="72" y="298" width="270" height="141">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="298" width="270" height="141">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="298" width="270" height="141"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="298" width="270" height="141">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="298" width="270" height="141"/>
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="298" width="270" height="141"/>
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="298" width="270" height="141"/>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="298" width="270" height="141">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="298" width="270" height="96">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="298" width="270" height="96"/>
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="“恒易贷”想访问您的通讯录" name="" label="“恒易贷”想访问您的通讯录" enabled="true" visible="true" x="88" y="317" width="238" height="21"/>
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="恒易贷需要获取您的通讯录信息，以提供更精准的服务" name="恒易贷需要获取您的通讯录信息，以提供更精准的服务" label="恒易贷需要获取您的通讯录信息，以提供更精准的服务" enabled="true" visible="true" x="88" y="341" width="238" height="33"/>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="394" width="270" height="1">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="394" width="270" height="1"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="394" width="270" height="1"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="394" width="270" height="1"/>
            </XCUIElementTypeOther>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="394" width="270" height="45">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="394" width="270" height="45"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="394" width="270" height="45"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="394" width="270" height="45"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="394" width="135" height="45">
                    <XCUIElementTypeButton type="XCUIElementTypeButton" name="不允许" label="不允许" enabled="true" visible="true" x="72" y="394" width="135" height="45"/>
                </XCUIElementTypeOther>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="207" y="394" width="1" height="45">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="207" y="394" width="1" height="45"/>
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="207" y="394" width="1" height="45"/>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    
```

```

        </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther">
enabled="true" visible="true" x="207" y="394" width="135" height="45">
            <XCUIElementTypeButton type="XCUIElementTypeButton">
                name="好" label="好" enabled="true" visible="true" x="207" y="394" width="135" height="45"/>
                    <XCUIElementTypeOther>
                    </XCUIElementTypeOther>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    </XCUIElementTypeOther>
</XCUIElementTypeAlert>
"""

accessYourP = re.compile("想访问您的") # “恒易贷”想访问您的通讯录
okChainList = [
{
    "tag": "XCUIElementTypeAlert",
    "attrs": {"enabled": "true", "visible": "true", "name": accessYourP}
},
{
    "tag": "XCUIElementTypeOther",
    "attrs": {"enabled": "true", "visible": "true"}
},
{
    "tag": "XCUIElementTypeButton",
    "attrs": {"enabled": "true", "visible": "true", "name": "好"}
}
]
okSoup = CommonUtils.bsChainFind(soup, okChainList)
if okSoup:
    # clickCenterPosition(curSession, okSoup.attrs)
    # foundAndProcessedPopup = True

    # # Note: seems actual click is No allow button? -> not work !
    # # change to wda query element then click for makesure work
    # curName = okSoup.attrs["name"] # 好
    # okButtonQuery = {"type": "XCUIElementTypeButton", "enabled": "true", "name": curName}
    #
    # foundAndClicked = self.findAndClickElement(okButtonQuery, isShowErrLog=False)
    # foundAndProcessedPopup = foundAndClicked
    foundAndProcessedPopup = self.findAndClickButtonElementBySoup(okSoup)
return foundAndProcessedPopup

```

iOSProcessCommonAlertAllow

```

def iOSProcessCommonAlertAllow(self, soup):
    foundAndProcessedPopup = False
    """

```

京东金融 弹框 想给您发送信息 允许:

```

<XCUIElementTypeAlert type="XCUIElementTypeAlert" name="" "京东金融"想给您发送通知
" label="" "京东金融"想给您发送通知" enabled="true" visible="true" x="72" y="298" width="270"
height="141">
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
x="72" y="298" width="270" height="141">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visib
le="true" x="72" y="298" width="270" height="141">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" v
isible="false" x="72" y="298" width="270" height="141">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true"
visible="false" x="72" y="298" width="270" height="141"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true"
visible="false" x="72" y="298" width="270" height="141"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled=
"true" visible="false" x="72" y="298" width="270" height="141"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled=
"true" visible="false" x="72" y="298" width="270" height="141"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled=
"true" visible="false" x="72" y="298" width="270" height="141"/>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" v
isible="true" x="72" y="298" width="270" height="141">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true"
visible="true" x="72" y="298" width="270" height="96">
                <XCUIElementTypeStaticText type="XCUIElementTypeStatic
Text" value="" "京东金融"想给您发送通知" name="" "京东金融"想给您发送通知" label="" "京东金融"想给您发
送通知" enabled="true" visible="true" x="88" y="317" width="238" height="21"/>
                <XCUIElementTypeStaticText type="XCUIElementTypeStatic
Text" value="" "通知"可能包括提醒、声音和图标标记。这些可在"设置"中配置。" name="" "通知"可能包括提醒
、声音和图标标记。这些可在"设置"中配置。" label="" "通知"可能包括提醒、声音和图标标记。这些可在"设置
"中配置。" enabled="true" visible="true" x="88" y="341" width="238" height="33"/>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true"
visible="true" x="72" y="394" width="270" height="1">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled=
"true" visible="false" x="72" y="394" width="270" height="1"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled=
"true" visible="true" x="72" y="394" width="270" height="1"/>
        </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true"
visible="true" x="72" y="394" width="270" height="45">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled=
"true" visible="true" x="72" y="394" width="270" height="45"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enab
led="true" visible="true" x="72" y="394" width="270" height="45">
                <XCUIElementTypeOther type="XCUIElementTypeOther"
enabled="true" visible="true" x="72" y="394" width="135" height="45">
                    <XCUIElementTypeButton type="XCUIElementTypeBu
tton" name="不允许" label="不允许" enabled="true" visible="true" x="72" y="394" width="135"
height="45"/>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    </XCUIElementTypeOther>
</XCUIElementTypeAlert>

```

```

        </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="207" y="394" width="1" height="45">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="207" y="394" width="1" height="45"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="207" y="394" width="1" height="45"/>
        </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="207" y="394" width="135" height="45">
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="允许" label="允许" enabled="true" visible="true" x="207" y="394" width="135" height="45"/>
        </XCUIElementTypeOther>
        </XCUIElementTypeOther>
        </XCUIElementTypeOther>
        </XCUIElementTypeOther>
        </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    </XCUIElementTypeAlert>

```

京东金融 弹框 允许“京东金融”在您使用该应用时访问您的位置吗？ 允许：

```

<XCUIElementTypeAlert type="XCUIElementTypeAlert" name="允许“京东金融”在您使用该应用时访问您的位置吗？" label="允许“京东金融”在您使用该应用时访问您的位置吗？" enabled="true" visible="true" x="72" y="247" width="270" height="243">
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="247" width="270" height="243">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="247" width="270" height="243">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="247" width="270" height="243">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="247" width="270" height="243"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="247" width="270" height="243">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="false" visible="false" x="72" y="247" width="270" height="243"/>
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="247" width="270" height="243"/>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="247" width="270" height="243">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="247" width="270" height="198">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="247" width="270" height="198">
                        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="允许“京东金融”在您使用该应用时访问您的位置吗？" name="允许“京东金融”在您使用该应用时访问您的位置吗？" label="允许“京东金融”在您使用该应用时访问您的位置吗？" enabled="true" visible="true" x="88" y="266" width="238" height="43"/>
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="允许“京东金融”在您使用该应用时访问您的位置吗？" name="允许“京东金融”在您使用该应用时访问您的位置吗？" label="允许“京东金融”在您使用该应用时访问您的位置吗？" enabled="true" visible="true" x="88" y="266" width="238" height="43"/>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    </XCUIElementTypeOther>

```

```

Text" value="为了基于您当前所在位置更精准地为您推荐附近的优惠、提供周边商家搜索服务、提供限定销售区域的金融产品，请您允许京东金融使用位置权限。我们仅定位您当前所处的位置，不会追踪您的行踪轨迹。您可以在设置页面取消位置授权。" name="为了基于您当前所在位置更精准地为您推荐附近的优惠、提供周边商家搜索服务、提供限定销售区域的金融产品，请您允许京东金融使用位置权限。我们仅定位您当前所处的位置，不会追踪您的行踪轨迹。您可以在设置页面取消位置授权。" label="为了基于您当前所在位置更精准地为您推荐附近的优惠、提供周边商家搜索服务、提供限定销售区域的金融产品，请您允许京东金融使用位置权限。我们仅定位您当前所处的位置，不会追踪您的行踪轨迹。您可以在设置页面取消位置授权。" enabled="true" visible="true"
" x="88" y="312" width="238" height="113"/>
    </XCUIElementTypeOther>
</XCUIElementTypeOther>
<XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="445" width="270" height="1">
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="445" width="270" height="1"/>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="445" width="270" height="1"/>
            </XCUIElementTypeOther>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="445" width="270" height="45">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="445" width="270" height="45">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="445" width="270" height="45">
                        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="445" width="135" height="45">
                            <XCUIElementTypeButton type="XCUIElementTypeButton" name="不允许" label="不允许" enabled="true" visible="true" x="72" y="445" width="135" height="45"/>
                            </XCUIElementTypeOther>
                            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="207" y="445" width="1" height="45">
                                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="207" y="445" width="1" height="45"/>
                                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="207" y="445" width="1" height="45"/>
                                    </XCUIElementTypeOther>
                                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="207" y="445" width="135" height="45">
                                        <XCUIElementTypeButton type="XCUIElementTypeButton" name="允许" label="允许" enabled="true" visible="true" x="207" y="445" width="135" height="45"/>
                                        </XCUIElementTypeOther>
                                        </XCUIElementTypeOther>
                                    </XCUIElementTypeOther>
                                </XCUIElementTypeOther>
                            </XCUIElementTypeOther>
                        </XCUIElementTypeOther>
                    </XCUIElementTypeOther>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    </XCUIElementTypeOther>
</XCUIElementTypeAlert>

```

京东金融 弹框 允许在您并未使用该应用时访问您的位置吗？仅在使用应用期间 始终允许：

```

<XCUIElementTypeAlert type="XCUIElementTypeAlert" name="允许“京东金融”在您并未使用该应用时访问您的位置吗？" label="允许“京东金融”在您并未使用该应用时访问您的位置吗？" enabled="t

```

```

    true" visible="true" x="72" y="224" width="270" height="288">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="224" width="270" height="288">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="224" width="270" height="288">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="224" width="270" height="288">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="224" width="270" height="288"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="224" width="270" height="288">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="224" width="270" height="288"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="224" width="270" height="288"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="224" width="270" height="288"/>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="224" width="270" height="288">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="224" width="270" height="199">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="224" width="270" height="199">
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="允许“京东金融”在您并未使用该应用时访问您的位置吗?" name="允许“京东金融”在您并未使用该应用时访问您的位置吗?" label="允许“京东金融”在您并未使用该应用时访问您的位置吗?" enabled="true" visible="true" x="88" y="244" width="238" height="43"/>
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="为了基于您当前所在位置更精准地为您推荐附近的优惠、提供周边商家搜索服务、提供限定销售区域的金融产品,请您允许京东金融使用位置权限。我们仅定位您当前所处的位置,不会追踪您的行踪轨迹。您可以在设置页面取消位置授权。" name="为了基于您当前所在位置更精准地为您推荐附近的优惠、提供周边商家搜索服务、提供限定销售区域的金融产品,请您允许京东金融使用位置权限。我们仅定位您当前所处的位置,不会追踪您的行踪轨迹。您可以在设置页面取消位置授权。" label="为了基于您当前所在位置更精准地为您推荐附近的优惠、提供周边商家搜索服务、提供限定销售区域的金融产品,请您允许京东金融使用位置权限。我们仅定位您当前所处的位置,不会追踪您的行踪轨迹。您可以在设置页面取消位置授权。" enabled="true" visible="true" x="88" y="290" width="238" height="112"/>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="422" width="270" height="1">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="422" width="270" height="1"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="422" width="270" height="1"/>
            </XCUIElementTypeOther>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="423" width="270" height="89">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="423" width="270" height="89"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="423" width="270" height="89"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="423" width="270" height="44"/>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    </XCUIElementTypeOther>

```

```

                <XCUIElementTypeButton type="XCUIElementTypeButton" name="仅在使用应用期间" label="仅在使用应用期间" enabled="true" visible="true" x="72" y="423" width="270" height="44"/>
                </XCUIElementTypeOther>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="467" width="270" height="1">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="467" width="270" height="1"/>
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="467" width="270" height="1"/>
                    </XCUIElementTypeOther>
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="467" width="270" height="45">
                        <XCUIElementTypeButton type="XCUIElementTypeButton" name="始终允许" label="始终允许" enabled="true" visible="true" x="72" y="467" width="270" height="270" height="45"/>
                        </XCUIElementTypeOther>
                    </XCUIElementTypeOther>
                    </XCUIElementTypeOther>
                    </XCUIElementTypeOther>
                    </XCUIElementTypeOther>
                </XCUIElementTypeAlert>

```

恒易贷 想给您发送通知 允许:

```

<XCUIElementTypeAlert type="XCUIElementTypeAlert" name="" "恒易贷"想给您发送通知" label="" "恒易贷"想给您发送通知" enabled="true" visible="true" x="72" y="298" width="270" height="141">
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="298" width="270" height="141">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="298" width="270" height="141">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="298" width="270" height="141">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="298" width="270" height="141"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="298" width="270" height="141"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="298" width="270" height="141"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="298" width="270" height="141"/>
            </XCUIElementTypeOther>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="298" width="270" height="141">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="298" width="270" height="96">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="298" width="270" height="96"/>
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="" "恒易贷"想给您发送通知" name="" "恒易贷"想给您发送通知" label="" "恒易贷"想给您发送通知"

```

```

    " enabled="true" visible="true" x="88" y="317" width="238" height="21"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="“通知”可能包括提醒、声音和图标标记。这些可在“设置”中配置。" name="“通知”可能包括提醒、声音和图标标记。这些可在“设置”中配置。" enabled="true" visible="true" x="88" y="341" width="238" height="33"/>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="394" width="270" height="1">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="394" width="270" height="1"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="394" width="270" height="1"/>
            </XCUIElementTypeOther>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="394" width="270" height="45">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="394" width="270" height="45">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="394" width="270" height="45">
                        <XCUIElementTypeButton type="XCUIElementTypeButton" name="不允许" label="不允许" enabled="true" visible="true" x="72" y="394" width="135" height="45"/>
                        </XCUIElementTypeOther>
                        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="207" y="394" width="1" height="45">
                            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="207" y="394" width="1" height="45"/>
                            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="207" y="394" width="1" height="45"/>
                            </XCUIElementTypeOther>
                            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="207" y="394" width="135" height="45">
                                <XCUIElementTypeButton type="XCUIElementTypeButton" name="允许" label="允许" enabled="true" visible="true" x="207" y="394" width="135" height="45"/>
                                </XCUIElementTypeOther>
                            </XCUIElementTypeOther>
                        </XCUIElementTypeOther>
                    </XCUIElementTypeOther>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    """
# allowP = re.compile("(始终)?允许") # will also match "不允许"
allowP = re.compile("^(始终)?允许$") # only match "允许" "始终允许"
allowChainList = [
{
    "tag": "XCUIElementTypeAlert",
    "attrs": {"enabled": "true", "visible": "true"}
}
]

```

```

    },
    {
        "tag": "XCUIElementTypeOther",
        "attrs": {"enabled":"true", "visible":"true"}
    },
    {
        "tag": "XCUIElementTypeButton",
        "# attrs": {"enabled":"true", "visible":"true", "name":"允许"}
        "# attrs": {"enabled":"true", "visible":"true", "name": allowP}
    },
],
allowSoup = CommonUtils.bsChainFind(soup, allowChainList)
if allowSoup:
    # clickCenterPosition(curSession, allowSoup.attrs)
    # foundAndProcessedPopup = True

    # # above click position not work for 允许 -> actually click 不允许 !!!
    # # change to use wda to find 允许 then click element
    # curName = allowSoup.attrs["name"] # 允许 / 始终允许
    # # allowButtonQuery = {"type":"XCUIElementTypeButton", "enabled":"true", "name":
"允许"}
    # allowButtonQuery = {"type":"XCUIElementTypeButton", "enabled":"true", "name": cu
rName}
    # foundAndClickeAllow = self.findAndClickElement(allowButtonQuery, isShowErrLog=Fa
lse)
    # foundAndProcessedPopup = foundAndClickeAllow
    foundAndProcessedPopup = self.findAndClickButtonElementBySoup(allowSoup)
return foundAndProcessedPopup

```



iOSProcessUserPrivacyProtocol

```

def iOSProcessUserPrivacyProtocol(self, soup):
    foundAndProcessedPopup = False
    """
        恒易贷 首次进入 用户隐私保护协议 点击 我接受
    """

    """
        恒易贷 用户意愿保护协议 我接收:
            <XCUIElementTypeWindow type="XCUIElementTypeWindow" enabled="true" visible="tr
ue" x="0" y="0" width="414" height="736">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
true" x="0" y="0" width="414" height="736">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visib
le="true" x="0" y="0" width="414" height="736">
                        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value=
"用户隐私保护协议" name="用户隐私保护协议" label="用户隐私保护协议" enabled="true" visib
le="true" x="93" y="86" width="228" height="34"/>
                        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value=
"感谢您使用恒易贷APP!" name="感谢您使用恒易贷APP!" label="感谢您使用恒易贷APP!" enab
led="true"
    
```

```

e" visible="true" x="35" y="143" width="156" height="24"/>
    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value=
"恒易贷非常重视您的隐私保护，并将使用多项安全防护措施来保护您的个人信息。请您在使用我们的服务前认真阅读本隐私政策，我们将通过本政策向您说明我们会如何收集、使用及保护您的个人信息。" name="恒易贷非常重视您的隐私保护，并将使用多项安全防护措施来保护您的个人信息。请您在使用我们的服务前认真阅读本隐私政策，我们将通过本政策向您说明我们会如何收集、使用及保护您的个人信息。" label="恒易贷非常重视您的隐私保护，并将使用多项安全防护措施来保护您的个人信息。请您在使用我们的服务前认真阅读本隐私政策，我们将通过本政策向您说明我们会如何收集、使用及保护您的个人信息。" enabled="true" visible="true" x="35" y
="178" width="344" height="86"/>
    <XCUIElementTypeOther type="XCUIElementTypeOther" name="点击我接受即表示您已阅读并同意《用户注册协议》和《用户隐私保护协议》，请详细阅读相关条款，如有疑问可拨打95713。" label="点击我接受即表示您已阅读并同意《用户注册协议》和《用户隐私保护协议》，请详细阅读相关条款，如有疑问可拨打95713。" enabled="true" visible="true" x="35" y="275" width="344" height="59"/>
    <XCUIElementTypeButton type="XCUIElementTypeButton" name="我接受" label="我接受" enabled="true" visible="true" x="67" y="393" width="280" height="44"/>
    <XCUIElementTypeButton type="XCUIElementTypeButton" name="我拒绝" label="我拒绝" enabled="true" visible="true" x="182" y="444" width="50" height="33"/>
</XCUIElementTypeOther>
</XCUIElementTypeOther>
</XCUIElementTypeWindow>
"""
userPrivacyIAcceptChainList = [
{
    "tag": "XCUIElementTypeWindow",
    "attrs": {"enabled": "true", "visible": "true", "x": "0", "y": "0", "width": "%s" % self.X, "height": "%s" % self.totalY}
},
{
    "tag": "XCUIElementTypeOther",
    "attrs": {"enabled": "true", "visible": "true", "x": "0", "y": "0", "width": "%s" % self.X, "height": "%s" % self.totalY}
},
{
    "tag": "XCUIElementTypeButton",
    "attrs": {"enabled": "true", "visible": "true", "name": "我接受"}
},
]
userPrivacyIAcceptSoup = CommonUtils.bsChainFind(soup, userPrivacyIAcceptChainList)
if userPrivacyIAcceptSoup:
    self.clickElementCenterPosition(userPrivacyIAcceptSoup)
    foundAndProcessedPopup = True
return foundAndProcessedPopup

```

iOSProcessCommonUserAgreementAgree

```

def iOSProcessCommonUserAgreementAgree(self, soup):
"""
    京东金融、必要 首次进入 授权协议提示 点击 同意
"""

```

```

foundAndProcessedAgreement = False
"""
京东金融 首次进入 协议提示 同意:
<XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
" x="0" y="0" width="414" height="736">
    ...
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
        x="0" y="579" width="414" height="157">
            <XCUIElementTypeOther type="XCUIElementTypeOther" name="有关个人信息收集
、使用更详细的约定请阅读《京东金融隐私政策》全文。我们承诺会不断完善安全技术和制度措施，确保您个人信
息的安全。" label="有关个人信息收集、使用更详细的约定请阅读《京东金融隐私政策》全文。我们承诺会不断
完善安全技术和制度措施，确保您个人信息的安全。" enabled="true" visible="true" x="16" y="595" wi
dth="382" height="59"/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="不同意" label
="不同意" enabled="true" visible="true" x="16" y="670" width="186" height="50"/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="同意" label="同
意" enabled="true" visible="true" x="212" y="670" width="186" height="50"/>
        </XCUIElementTypeOther>
    </XCUIElementTypeOther>
</XCUIElementTypeOther>

必要 用户服务协议及必要隐私政策 同意:
<XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
" x="0" y="0" width="414" height="736">
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
    x="38" y="163" width="338" height="410">
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="用户
服务协议及必要隐私政策" name="用户服务协议及必要隐私政策" label="用户服务协议及必要隐私政策" enab
led="true" visible="true" x="38" y="163" width="338" height="53"/>
        <XCUIElementTypeTextView type="XCUIElementTypeTextView" value="在您注册
成为必要用户的进程中，您需要完成我们的注册流程并通过点击同意的形式在线签署以下协议，请您务必仔细阅读
、充分理解协议中的条款内容后再点击同意（尤其是以粗体标识的条款，因为这些条款可能会明确您应履行的义务
或对您的权利有所限制）：&#10;《用户服务协议》&#10;《必要隐私政策》&#10;【请您注意】如果您不同意上
述协议或其中任何条款约定，请您停止注册。您停止注册后将仅可以浏览我们的商品信息但无法享受我们的产品或
服务。如您按照注册流程提示填写信息、阅读并点击同意上述协议且完成全部注册流程后，即表示您已充分阅读、
理解并接受协议的全部内容；并表明您也同意必要可以依据以上的隐私政策内容来处理您的个人信息。如您对以上
协议内容有任何疑问，您可随时与必要客服联系。" enabled="true" visible="true" x="60" y="215" wid
th="294" height="241"/>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visib
le="true" x="60" y="455" width="294" height="2"/>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visib
le="true" x="60" y="467" width="294" height="56"/>
        <XCUIElementTypeButton type="XCUIElementTypeButton" name="不同意" label
="不同意" enabled="true" visible="true" x="38" y="518" width="169" height="55"/>
        <XCUIElementTypeButton type="XCUIElementTypeButton" name="同意" label="同
意" enabled="true" visible="true" x="206" y="518" width="170" height="55"/>
    </XCUIElementTypeOther>
</XCUIElementTypeOther>

斑马AI课 弹框 个人信息保护政策 同意:
<XCUIElementTypeWindow type="XCUIElementTypeWindow" enabled="true" visible="tr
ue" x="0" y="0" width="414" height="736">
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
x="0" y="0" width="414" height="736">

```

```

        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="0" width="414" height="736"/>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="0" width="414" height="736"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="47" y="148" width="320" height="440">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="个人信息保护政策" name="个人信息保护政策" label="个人信息保护政策" enabled="true" visible="true" x="47" y="173" width="320" height="23"/>
                <XCUIElementTypeTextView type="XCUIElementTypeTextView" value="感谢您信任并使用斑马AI课的产品和服务。我们根据最新的法律法规、监管政策要求，更新了《隐私政策》，特向您推送本提示。请您仔细阅读并充分理解相关条款。&#10;斑马AI课会通过《隐私政策》帮助您了解我们手机、使用、存储个人信息的情况，以及您享有的相关权利。&#10;点击【同意】，表示您已阅读并同意相关协议条款，斑马AI课将尽全力保障您的合法权益并继续为您提供优质的产品和服务。&#10;* 您可通过阅读完整版《用户注册协议》《用户隐私协议》《儿童隐私政策》了解详尽条款内容。" enabled="true" visible="true" x="67" y="210" width="280" height="301">
                    <XCUIElementTypeLink type="XCUIElementTypeLink" name="《用户注册协议》" label="《用户注册协议》" enabled="true" visible="true" x="210" y="470" width="115" height="25"/>
                    <XCUIElementTypeLink type="XCUIElementTypeLink" name="《用户隐私协议》" label="《用户隐私协议》" enabled="true" visible="true" x="72" y="493" width="115" height="25"/>
                    <XCUIElementTypeLink type="XCUIElementTypeLink" name="《儿童隐私政策》" label="《儿童隐私政策》" enabled="true" visible="true" x="186" y="493" width="115" height="25"/>
                </XCUIElementTypeTextView>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="47" y="525" width="320" height="2"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="同意" label="同意" enabled="true" visible="true" x="77" y="540" width="260" height="34"/>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    </XCUIElementTypeWindow>
"""

commonAgreeChainList = [
    {
        "tag": "XCUIElementTypeOther",
        "attrs": {"enabled": "true", "visible": "true", "x": "0", "y": "0", "width": "%s" % self.X, "height": "%s" % self.totalY}
    },
    {
        "tag": "XCUIElementTypeOther",
        # "attrs": {"enabled": "true", "visible": "true", "x": "0", "width": "%s" % self.X}

        "attrs": {"enabled": "true", "visible": "true"}
    },
    {
        "tag": "XCUIElementTypeButton",
        "attrs": {"enabled": "true", "visible": "true", "name": "同意"}
    },
]
commonAgreeSoup = CommonUtils.bsChainFind(soup, commonAgreeChainList)
if commonAgreeSoup:

```

```

    self.clickElementCenterPosition(commonAgreeSoup)
    foundAndProcessedAgreement = True
    return foundAndProcessedAgreement

```

iOSProcessPopupIKnow

```

def iOSProcessPopupIKnow(self, soup):
    """
        京东金融 首次打开 协议弹框 我知道了
    """
    foundAndProcessedPopup = False
    """
        京东金融 弹框 我知道了:
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
        " x="0" y="0" width="414" height="736">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
        true" x="67" y="189" width="280" height="358">
                <XCUIElementTypeScrollView type="XCUIElementTypeScrollView" enabled="t
        rue" visible="true" x="67" y="217" width="280" height="280">
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value=
        "权限调用说明" name="权限调用说明" label="权限调用说明" enabled="true" visible="true" x="77" y
        ="217" width="260" height="21"/>
                    <XCUIElementTypeOther type="XCUIElementTypeOther" name="在提供服务过
        程中, 我们可能需要调用您的以下重要设备权限, 我们将在首次调用时逐项询问您是否允许使用该权限。您可以在
        我们询问时开启相关权限, 也可以在设备系统“设置”里管理相关权限: &#10;1、相机、相册权限: 向您提供扫一扫、头像设置、客服、评论或分享、人脸等图像识别服务时, 您可以通过开通相机权限和/或相册权限以便进行实
        时拍摄和图片/视频上传。&#10;2、位置权限: 基于您当前位置为您自动定位商家网点(银行、加油站等)、推荐周
        边服务及优惠、提供限定销售区域的金融服务时, 您可以通过开启位置权限以便查看或获取当前所在区域的服务
        。&#10;3、通讯录权限: 向您提供手机充值、转账、还款服务时, 您可以选择直接从通讯录中选择并导入选定的
        特定联系人信息。&#10;4、日历权限: 向您提供早起打卡提醒服务时, 您可以通过开启日历权限便捷管理您的自
        定义事项、设定重要事项提醒。&#10;5、麦克风权限: 向您提供语音搜索或语音客服服务时, 您可以开启麦克风
        权限点击语音按钮进行录音, 以便我们收集您的语音内容并进行必要的处理。&#10;6、面容 ID权限: 向您提供面
        容 ID支付、面容 ID解锁服务时, 您可以开启面容 ID权限并在您的设备上完成面容 ID验证, 以便我们接收使用
        验证结果。&#10;7、蓝牙权限: 向您提供手环服务时, 您可以开启蓝牙权限搜索周边设备, 以便您的手环能够与
        京东金融APP建立正常连接。&#10;&#10;更多权限信息说明, 请点击查看完整版《京东金融隐私政策》。&#10;&
        10;" label="在提供服务过程中, 我们可能需要调用您的以下重要设备权限, 我们将在首次调用时逐项询问您是
        否允许使用该权限。您可以在我们询问时开启相关权限, 也可以在设备系统“设置”里管理相关权限: &#10;1、相
        机、相册权限: 向您提供扫一扫、头像设置、客服、评论或分享、人脸等图像识别服务时, 您可以通过开通相机权
        限和/或相册权限以便进行实时拍摄和图片/视频上传。&#10;2、位置权限: 基于您当前位置为您自动定位商家网
        点(银行、加油站等)、推荐周边服务及优惠、提供限定销售区域的金融服务时, 您可以通过开启位置权限以便查
        看或获取当前所在区域的服务。&#10;3、通讯录权限: 向您提供手机充值、转账、还款服务时, 您可以选择直接从通
        讯录中选择并导入选定的特定联系人信息。&#10;4、日历权限: 向您提供早起打卡提醒服务时, 您可以通过开启日
        历权限便捷管理您的自定义事项、设定重要事项提醒。&#10;5、麦克风权限: 向您提供语音搜索或语音客服服务时,
        您可以开启麦克风权限点击语音按钮进行录音, 以便我们收集您的语音内容并进行必要的处理。&#10;
        6、面容 ID权限: 向您提供面容 ID支付、面容 ID解锁服务时, 您可以开启面容 ID权限并在您的设备上完成面
        容 ID验证, 以便我们接收使用验证结果。&#10;7、蓝牙权限: 向您提供手环服务时, 您可以开启蓝牙权限搜索周
        边设备, 以便您的手环能够与京东金融APP建立正常连接。&#10;&#10;更多权限信息说明, 请点击查看完整版《
        京东金融隐私政策》。&#10;&#10;" enabled="true" visible="true" x="77" y="257" width="261" hei
        ght="888"/>
            </XCUIElementTypeScrollView>
    
```

```

        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="67" y="497" width="280" height="1"/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="我知道了" label="我知道了" enabled="true" visible="true" x="67" y="497" width="280" height="50"/>
        </XCUIElementTypeOther>
    </XCUIElementTypeOther>

    恒易贷 权限获取说明 我知道了:
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="736">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="736"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="37" y="151" width="340" height="434">
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="权限获取说明" name="权限获取说明" label="权限获取说明" enabled="true" visible="true" x="158" y="171" width="98" height="20"/>
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="为了给您提供优质的服务,在您借款过程中,恒易贷需要您授权开启以下权限: " name="为了给您提供优质的服务,在您借款过程中,恒易贷需要您授权开启以下权限: " label="为了给您提供优质的服务,在您借款过程中,恒易贷需要您授权开启以下权限: " enabled="true" visible="true" x="62" y="196" width="290" height="33"/>
                    <XCUIElementTypeTable type="XCUIElementTypeTable" enabled="true" visible="true" x="62" y="246" width="290" height="274">
                        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="62" y="246" width="290" height="1"/>
                            <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="62" y="246" width="290" height="55">
                                <XCUIElementTypeImage type="XCUIElementTypeImage" name="icon_notice" enabled="true" visible="false" x="62" y="246" width="17" height="19"/>
                                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="通知权限" name="通知权限" label="通知权限" enabled="true" visible="true" x="87" y="246" width="62" height="24"/>
                                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="用于为您提供恒易贷或合作方的产品或活动信息" name="用于为您提供恒易贷或合作方的产品或活动信息" label="用于为您提供恒易贷或合作方的产品或活动信息" enabled="true" visible="true" x="87" y="271" width="265" height="19"/>
                            </XCUIElementTypeCell>
                            <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="62" y="300" width="290" height="73">
                                <XCUIElementTypeImage type="XCUIElementTypeImage" name="icon_location" enabled="true" visible="false" x="62" y="300" width="17" height="19"/>
                                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="定位服务权限" name="定位服务权限" label="定位服务权限" enabled="true" visible="true" x="87" y="300" width="92" height="24"/>
                                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="用于获取您的地理位置信息,以为您提供匹配的风险信资评估服务" name="用于获取您的地理位置信息,以为您提供匹配的风险信资评估服务" label="用于获取您的地理位置信息,以为您提供匹配的风险信资评估服务" enabled="true" visible="true" x="87" y="325" width="265" height="33"/>
                            </XCUIElementTypeCell>
                            <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="62" y="372" width="290" height="73">
                                <XCUIElementTypeImage type="XCUIElementTypeImage" name="icon_m

```

```

        ail" enabled="true" visible="false" x="62" y="372" width="17" height="19"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="通讯录权限" name="通讯录权限" label="通讯录权限" enabled="true" visible="true" x="87" y="372" width="77" height="24"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="用于调用设备通讯录以方便您选择联系人电话号码, 以辅助填写信息" name="用于调用设备通讯录以方便您选择联系人电话号码, 以辅助填写信息" label="用于调用设备通讯录以方便您选择联系人电话号码, 以辅助填写信息" enabled="true" visible="true" x="87" y="397" width="265" height="33"/>
        </XCUIElementTypeCell>
        <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="62" y="444" width="290" height="73">
            <XCUIElementTypeImage type="XCUIElementTypeImage" name="icon_camera" enabled="true" visible="false" x="62" y="444" width="17" height="19"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="相机权限" name="相机权限" label="相机权限" enabled="true" visible="true" x="87" y="444" width="62" height="24"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="用于获取您的身份证件信息以及面部图像信息, 以核实您身份的真实性" name="用于获取您的身份证件信息以及面部图像信息, 以核实您身份的真实性" label="用于获取您的身份证件信息以及面部图像信息, 以核实您身份的真实性" enabled="true" visible="true" x="87" y="469" width="265" height="33"/>
        </XCUIElementTypeCell>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="62" y="516" width="290" height="1"/>
    </XCUIElementTypeTable>
    <XCUIElementTypeButton type="XCUIElementTypeButton" name="我知道了" label="我知道了" enabled="true" visible="true" x="92" y="520" width="230" height="38"/>
        </XCUIElementTypeOther>
    </XCUIElementTypeOther>
    """
    iKnowChainList = [
        {
            "tag": "XCUIElementTypeOther",
            "attrs": {"enabled": "true", "visible": "true", "x": "0", "y": "0", "width": "%s" % self.X, "height": "%s" % self.totally}
        },
        {
            "tag": "XCUIElementTypeOther",
            "attrs": {"enabled": "true", "visible": "true"}
        },
        {
            "tag": "XCUIElementTypeButton",
            "attrs": {"enabled": "true", "visible": "true", "name": "我知道了"}
        },
    ]
    iKnowSoup = CommonUtils.bsChainFind(soup, iKnowChainList)
    if iKnowSoup:
        self.clickElementCenterPosition(iKnowSoup)
        foundAndProcessedPopup = True
    return foundAndProcessedPopup

```



iOSProcessPopupPhoneCall


```

<XCUIElementTypeOther type="XCUIElementTypeOther"
enabled="true" visible="true" x="207" y="376" width="1" height="45">
    <XCUIElementTypeOther type="XCUIElementTypeOther"
enabled="true" visible="false" x="207" y="376" width="1" height="45"/>
    <XCUIElementTypeOther type="XCUIElementTypeOther"
enabled="true" visible="true" x="207" y="376" width="1" height="45"/>
</XCUIElementTypeOther>
<XCUIElementTypeOther type="XCUIElementTypeOther"
enabled="true" visible="true" x="207" y="376" width="135" height="45">
    <XCUIElementTypeButton type="XCUIElementTypeButton"
name="呼叫" label="呼叫" enabled="true" visible="true" x="207" y="376" width="135" height="45"/>
        </XCUIElementTypeOther>
        </XCUIElementTypeOther>
        </XCUIElementTypeOther>
        </XCUIElementTypeOther>
        </XCUIElementTypeOther>
        </XCUIElementTypeOther>
        </XCUIElementTypeOther>
</XCUIElementTypeOther>
</XCUIElementTypeAlert>
"""
callChainList = [
{
    "tag": "XCUIElementTypeAlert",
    "attrs": {"enabled": "true", "visible": "true"}
},
{
    "tag": "XCUIElementTypeOther",
    "attrs": {"enabled": "true", "visible": "true"}
},
{
    "tag": "XCUIElementTypeButton",
    "attrs": {"enabled": "true", "visible": "true", "name": "呼叫"}
},
]
callSoup = CommonUtils.bsChainFind(soup, callChainList)
if callSoup:
    # parentOtherSoup = callSoup.parent
    # if parentOtherSoup:
    #     parentParentOtherSoup = parentOtherSoup.parent
    #     if parentParentOtherSoup:
    #         cancelSoup = parentParentOtherSoup.find(
    #             "XCUIElementTypeButton",
    #             attrs={"enabled": "true", "visible": "true", "name": "取消"})
    #         if cancelSoup:
    #             clickCenterPosition(curSession, cancelSoup.attrs)
    #             foundAndProcessedPopup = True

    # # above click position not work for 取消 !!!
    # # change to find 取消 then click element
    # cancelButtonQuery = {"type": "XCUIElementTypeButton", "enabled": "true", "visible": "true", "name": "取消"}]
```

```

        # foundAndClickCancelButton = self.findAndClickElement(cancelButtonQuery, isShowErrLog=False)
        # foundAndProcessedPopup = foundAndClickCancelButton
        foundAndProcessedPopup = self.findAndClickButtonElementBySoup(curButtonName="取消")
        return foundAndProcessedPopup
    
```

iOSProcessCommonPopupWindowUpperRightClose

```

# 注: 此逻辑经常误判, 没有弹框误以为有弹框, 所以放弃此逻辑
def iOSProcessCommonPopupWindowUpperRightClose(self, soup):
    """iOS 常见 通用弹框: 中间有多层 XCUIElementTypeOther 且 是全屏大小
    enabled="true" visible="true" x="0" y="0" width="414" height="736"
    至少3层, 其下 再去找 就是 弹框本身最外层元素了
    然后尝试点击右上角关闭按钮
    """
    foundAndProcessedPopup = False
    """
    弹框中找 外层Other内部的 非x=0 y=0的元素, 则为弹框区域 计算出其右上角的区域, 点击 尝试关闭 弹
    框
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
    " x="0" y="0" width="414" height="736">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
        true" x="0" y="0" width="414" height="736">
            <XCUIElementTypeOther type="XCUIElementTypeOther" name="姜老师 姜老师已
            经帮助超过800多名患者发起了筹款, 经验丰富。关于筹款的问题您都可以找她咨询。 gongyiwuyouchou 一键复
            制老师微信 添加老师微信, 专享1对1细致服务, 立即开始咨询吧! " label="姜老师 姜老师已经帮助超过800多
            名患者发起了筹款, 经验丰富。关于筹款的问题您都可以找她咨询。 gongyiwuyouchou 一键复制老师微信 添
            加老师微信, 专享1对1细致服务, 立即开始咨询吧! " enabled="true" visible="true" x="0" y="0" width=
            "414" height="736">
                <XCUIElementTypeOther type="XCUIElementTypeOther" name="姜老师 姜老
                师已经帮助超过800多名患者发起了筹款, 经验丰富。关于筹款的问题您都可以找她咨询。 gongyiwuyouchou
                一键复制老师微信 添加老师微信, 专享1对1细致服务, 立即开始咨询吧! " label="姜老师 姜老师已经帮助超
                过800多名患者发起了筹款, 经验丰富。关于筹款的问题您都可以找她咨询。 gongyiwuyouchou 一键复制老师
                微信 添加老师微信, 专享1对1细致服务, 立即开始咨询吧! " enabled="true" visible="true" x="0" y="0" wid
                th="414" height="736">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" name="姜老师 姜老
                    师已经帮助超过800多名患者发起了筹款, 经验丰富。关于筹款的问题您都可以找她咨询。 gongyiwuyouchou
                    一键复制老师微信 添加老师微信, 专享1对1细致服务, 立即开始咨询吧! " label="姜老师 姜老师已经帮助超
                    过800多名患者发起了筹款, 经验丰富。关于筹款的问题您都可以找她咨询。 gongyiwuyouchou 一键复制老师
                    微信 添加老师微信, 专享1对1细致服务, 立即开始咨询吧! " enabled="true" visible="true" x="20" y="200" width=
                    "374" height="263">
                        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled=
                        "true" visible="true" x="20" y="200" width="374" height="36"/>
                        ...
    """
    allTopOtherSoupList = soup.findall(
        "XCUIElementTypeOther",
        attrs={"enabled": "true", "visible": "true", "x": "0", "y": "0", "width": "%s" % self.X,
        "height": "%s" % self.totalY}
    )
    for eachTopOtherSoup in allTopOtherSoupList:
        if eachTopOtherSoup.get("name") == "取消":
            self.click_element(eachTopOtherSoup)
            break
    else:
        self.click_element(foundAndProcessedPopup)

```

```

popupTopOtherSoup = self.findPopupTopFrameElement(eachTopOtherSoup)
if popupTopOtherSoup:
    curAttrsDict = popupTopOtherSoup.attrs
    curX = int(curAttrsDict["x"])
    curY = int(curAttrsDict["y"])
    curWidth = int(curAttrsDict["width"])
    curHeight = int(curAttrsDict["height"])
    curX1 = curX + curWidth
    curY1 = curY + curHeight
    PossibleCloseButtonWidth = 40
    PossibleCloseButtonHeight = 40
    possibleCloseButtonX0 = curX1 - PossibleCloseButtonWidth
    # possibleCloseButtonY0 = curY + PossibleCloseButtonHeight
    possibleCloseButtonY0 = curY
    possibleCloseButtonX1 = curX1
    possibleCloseButtonY1 = curY + PossibleCloseButtonHeight
    # possibleCloseButtonCenterX = possibleCloseButtonX0 + int(PossibleCloseButton
Width / 2)
    # possibleCloseButtonCenterY = possibleCloseButtonY0 + int(PossibleCloseButton
Height / 2)
    possibleCloseButtonCenterX = int((possibleCloseButtonX0 + possibleCloseButton
X1) / 2)
    possibleCloseButtonCenterY = int((possibleCloseButtonY0 + possibleCloseButton
Y1) / 2)
    possibleCloseButtonCenterPositon = (possibleCloseButtonCenterX, possibleCloseB
uttonCenterY)
    self.tap(possibleCloseButtonCenterPositon)
    foundAndProcessedPopup = True
    break

return foundAndProcessedPopup

```

findPopupTopFrameElement

```

def findPopupTopFrameElement(self, topOtherSoup):
    """
    寻找符合条件的子节点，即当前节点向下找，符合一直是
    type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414"
    height="736"
    且层数 >= 3，然后再找其下一个 非x=0 y=0的节点
    很可能就是 弹框的主体元素
    """
    popupTopFrameElement = None

    # MaxFullScreenSizeLevel = 3
    # FullScreenSizeMinLevel = 3
    # FullScreenSizeMaxLevel = 6 # see many invalid case is level >= 10, so here restrict
    to 6, not too much level
    FullScreenSizLevel = 3 # special: 无忧筹 点击 专属老师 后的弹框

```

```

FullScreenAttr = {"type": "XCUIElementTypeOther", "enabled": "true", "visible": "true",
"x": "0", "y": "0", "width": "%s" % self.X, "height": "%s" % self.totalY}

curFullScreenOtherLevel = 0
curSoup = topOtherSoup
while True:
    subOtherSoupList = curSoup.find_all(
        "XCUIElementTypeOther",
        attrs=FullScreenAttr,
        recursive=False,
    )

    if not subOtherSoupList:
        break

    # only have one child
    childOtherSoupNum = len(subOtherSoupList)
    if childOtherSoupNum != 1:
        break

    firstOnlyOtherSoup = subOtherSoupList[0]
    if not firstOnlyOtherSoup:
        break

    if not hasattr(firstOnlyOtherSoup, "attrs"):
        break

    curAttrDict = firstOnlyOtherSoup.attrs

    allAttrSame = True
    for eachToCompareKey in FullScreenAttr.keys():
        eachToCompareValue = FullScreenAttr[eachToCompareKey]
        curValue = curAttrDict.get(eachToCompareKey)
        if curValue != eachToCompareValue:
            allAttrSame = False
            break

    if not allAttrSame:
        break

    curSoup = firstOnlyOtherSoup

    curFullScreenOtherLevel += 1

    # if curFullScreenOtherLevel >= MaxFullScreenSizeLevel:
    #     isPossiblePopup = FullScreenSizeMinLevel <= curFullScreenOtherLevel < FullScreenSize
    #     MaxLevel
    isPossiblePopup = curFullScreenOtherLevel == FullScreenSizeLevel
    if isPossiblePopup:
        curChildOtherList = curSoup.find_all(
            "XCUIElementTypeOther",
            attrs={"type": "XCUIElementTypeOther", "enabled": "true", "visible": "true"},
            recursive=False,
        )

```

```

        )
    if curChildOtherList:
        curChildOtherNum = len(curChildOtherList)
        if curChildOtherNum == 1:
            popupTopFrameElement = curChildOtherList[0]

    return popupTopFrameElement

```

iOSProcessPopupPhotoCamera

```

def iOSProcessPopupPhotoCamera(self, soup):
    """ios 系统弹框： 拍照或录像 照片图库 浏览 取消"""
    foundAndProcessedPopup = False
    """
        系统弹框 拍照或录像：
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
    " x="8" y="530" width="398" height="133">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
    true" x="8" y="530" width="398" height="133">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visib
    le="true" x="8" y="530" width="398" height="133">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" v
    isible="false" x="0" y="0" width="398" height="133">
                        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="tr
    ue" visible="true" x="0" y="0" width="398" height="133">
                            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled=
    "true" visible="true" x="0" y="0" width="398" height="133">
                                <XCUIElementTypeOther type="XCUIElementTypeOther" enab
    led="true" visible="true" x="0" y="0" width="398" height="133">
                                    <XCUIElementTypeOther type="XCUIElementTypeOther"
    enabled="true" visible="true" x="0" y="0" width="398" height="133">
                                        <XCUIElementTypeTable type="XCUIElementType
    Table" enabled="true" visible="true" x="0" y="0" width="398" height="133">
                                            <XCUIElementTypeCell type="XCUIElement
    TypeCell" enabled="true" visible="true" x="0" y="0" width="398" height="45">
                                                <XCUIElementTypeOther type="XCUIEl
    ementTypeOther" enabled="true" visible="true" x="0" y="43" width="398" height="2"/>
                                                <XCUIElementTypeImage type="XCUIEl
    ementTypeImage" enabled="true" visible="false" x="13" y="6" width="1" height="32"/>
                                                <XCUIElementTypeStaticText type="X
    CUIElementTypeStaticText" value="拍照或录像" name="拍照或录像" label="拍照或录像" enab
    led="true" visible="true" x="15" y="11" width="87" height="22"/>
                                                <XCUIElementTypeImage type="XCUIEl
    ementTypeImage" enabled="true" visible="false" x="351" y="6" width="32" height="32"/>
                                            </XCUIElementTypeCell>
                                            <XCUIElementTypeCell type="XCUIElement
    TypeCell" enabled="true" visible="true" x="0" y="44" width="398" height="45">
                                                <XCUIElementTypeOther type="XCUIEl
    ementTypeOther" enabled="true" visible="true" x="0" y="88" width="398" height="1"/>

```



```

photoCameraSoup = CommonUtils.bsChainFind(soup, photoCameraChainList)
if photoCameraSoup:
    topParentOtherSoup = None
    # find top most parent
    parentLevelNum = 11
    curParentSoup = photoCameraSoup
    for curLevelIdx in range(parentLevelNum):
        curParentSoup = curParentSoup.parent
        if not curParentSoup:
            break
    if curParentSoup:
        topParentOtherSoup = curParentSoup

    if topParentOtherSoup:
        nextSiblingList = topParentOtherSoup.next_siblings
        if nextSiblingList:
            nextOtherSoup = None
            for eachNextSibling in nextSiblingList:
                if hasattr(eachNextSibling, "attrs"):
                    curType = eachNextSibling.attrs["type"]
                    if curType == "XCUIElementTypeOther":
                        # next sibling's first XCUIElementTypeOther
                        nextOtherSoup = eachNextSibling
                        break

            if nextOtherSoup:
                cancelSoup = nextOtherSoup.find(
                    "XCUIElementTypeButton",
                    attrs={"visible": "true", "enabled": "true", "name": "取消"})
                if cancelSoup:
                    self.clickElementCenterPosition(cancelSoup)
                    foundAndProcessedPopup = True

return foundAndProcessedPopup

```

iOSProcessAlertNoteConfrim

```

def iOSProcessAlertNoteConfrim(self, soup):
    """只要符合XCUIElementTypeAlert 且其中有 确定 按钮，则点击"""
    foundAndProcessedPopup = False
    """
        康爱公社 弹框 提醒 您的身份信息不完整 确定：
        <XCUIElementTypeAlert type="XCUIElementTypeAlert" name="提醒" label="提醒" enabled="true" visible="true" x="72" y="298" width="270" height="141">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="298" width="270" height="141">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="298" width="270" height="141">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" v

```

```

    visible="false" x="72" y="298" width="270" height="141">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="298" width="270" height="141"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="298" width="270" height="141">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="298" width="270" height="141"/>
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="298" width="270" height="141"/>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="298" width="270" height="141">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="298" width="270" height="96">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="298" width="270" height="96">
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="提醒" name="提醒" label="提醒" enabled="true" visible="true" x="88" y="317" width="238" height="21"/>
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="您的身份信息不完整,请检查身份证信息及手机号是否完善" name="您的身份信息不完整,请检查身份证信息及手机号是否完善" label="您的身份信息不完整,请检查身份证信息及手机号是否完善" enabled="true" visible="true" x="88" y="341" width="238" height="33"/>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="394" width="270" height="1">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="72" y="394" width="270" height="1"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="394" width="270" height="1"/>
            </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="394" width="270" height="45">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="394" width="270" height="45">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="72" y="394" width="270" height="45">
                    <XCUIElementTypeButton type="XCUIElementTypeButton" name="确定" label="确定" enabled="true" visible="true" x="72" y="394" width="270" height="45"/>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    </XCUIElementTypeOther>
</XCUIElementTypeAlert>

```



```

        </XCUIElementTypeOther>
    </XCUIElementTypeOther>
    </XCUIElementTypeOther>
    </XCUIElementTypeOther>
</XCUIElementTypeAlert>
"""

noteTipP = re.compile("((提醒)|(提示))")
alertConfirmChainList = [
    {
        "tag": "XCUIElementTypeAlert",
        # "attrs": {"enabled": "true", "visible": "true"}
        "attrs": {"enabled": "true", "visible": "true", "name": noteTipP}
    },
    {
        "tag": "XCUIElementTypeOther",
        "attrs": {"enabled": "true", "visible": "true"}
    },
    {
        "tag": "XCUIElementTypeButton",
        "attrs": {"enabled": "true", "visible": "true", "name": "确定"}
    }
]
alertConfirmSoup = CommonUtils.bsChainFind(soup, alertConfirmChainList)
if alertConfirmSoup:
    self.clickElementCenterPosition(alertConfirmSoup)
    foundAndProcessedPopup = True
return foundAndProcessedPopup

```



iOSProcessAlertCancel

```

def iOSProcessAlertCancel(self, soup):
    """对于常见的XCUIElementTypeAlert，内有 取消，则点击取消"""
    foundAndProcessedPopup = False
    """
    益路通行 弹框 您是否要退出登录：
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
    " x="0" y="0" width="414" height="736">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
true" x="0" y="0" width="414" height="736"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
false" x="0" y="0" width="414" height="736"/>
                    <XCUIElementTypeAlert type="XCUIElementTypeAlert" name="您是否要退出登录?" 
label="您是否要退出登录?" enabled="true" visible="true" x="72" y="316" width="270" height="
105">
                        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visib
le="true" x="72" y="316" width="270" height="105">
                            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" v
isible="true" x="72" y="316" width="270" height="105">
                                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="tru
e" visible="true" x="72" y="316" width="270" height="105">

```

```

e" visible="false" x="72" y="316" width="270" height="105">
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled=
"true" visible="false" x="72" y="316" width="270" height="105"/>
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled=
"true" visible="false" x="72" y="316" width="270" height="105">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enab
led="true" visible="false" x="72" y="316" width="270" height="105"/>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enab
led="true" visible="false" x="72" y="316" width="270" height="105"/>
    </XCUIElementTypeOther>
</XCUIElementTypeOther>
<XCUIElementTypeOther type="XCUIElementTypeOther" enabled="tru
e" visible="true" x="72" y="316" width="270" height="105">
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled=
"true" visible="true" x="72" y="316" width="270" height="60">
        <XCUIElementTypeOther type="XCUIElementTypeStaticText" type="XCUIElementTypeSt
aticText" value="您是否要退出登录?" name="您是否要退出登录?" label="您是否要退出登录?" enabl
ed="true" visible="true" x="88" y="335" width="238" height="21"/>
    </XCUIElementTypeOther>
</XCUIElementTypeOther>
<XCUIElementTypeOther type="XCUIElementTypeOther" enabled=
"true" visible="true" x="72" y="376" width="270" height="1">
    <XCUIElementTypeOther type="XCUIElementTypeOther" enab
led="true" visible="false" x="72" y="376" width="270" height="1"/>
    <XCUIElementTypeOther type="XCUIElementTypeOther" enab
led="true" visible="true" x="72" y="376" width="270" height="1"/>
    </XCUIElementTypeOther>
<XCUIElementTypeOther type="XCUIElementTypeOther" enabled=
"true" visible="true" x="72" y="376" width="270" height="45">
    <XCUIElementTypeOther type="XCUIElementTypeOther" enab
led="true" visible="true" x="72" y="376" width="270" height="45">
        <XCUIElementTypeOther type="XCUIElementTypeOther" type="XCUIElementTypeOther"
enabled="true" visible="true" x="72" y="376" width="135" height="45">
            <XCUIElementTypeButton type="XCUIElementTypeButt
on" name="取消" label="取消" enabled="true" visible="true" x="72" y="376" width="135"
height="45"/>
        </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled=
"true" visible="true" x="207" y="376" width="1" height="45">
            <XCUIElementTypeOther type="XCUIElementTypeTyp
eOther" enabled="true" visible="false" x="207" y="376" width="1" height="45"/>
            <XCUIElementTypeOther type="XCUIElementTypeTyp
eOther" enabled="true" visible="true" x="207" y="376" width="1" height="45"/>
        </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled=
"true" visible="true" x="207" y="376" width="135" height="45">
            <XCUIElementTypeButton type="XCUIElementTypeButt
on" name="退出" label="退出" enabled="true" visible="true" x="207" y="376" width="135"
height="45"/>
        </XCUIElementTypeOther>
    </XCUIElementTypeOther>
</XCUIElementTypeOther>

```

```

                </XCUIElementTypeOther>
                </XCUIElementTypeOther>
                </XCUIElementTypeOther>
                </XCUIElementTypeOther>
                </XCUIElementTypeOther>
                </XCUIElementTypeAlert>
            """
        alertCancelChainList = [
            {
                "tag": "XCUIElementTypeAlert",
                "attrs": {"enabled": "true", "visible": "true"}
            },
            {
                "tag": "XCUIElementTypeOther",
                "attrs": {"enabled": "true", "visible": "true"}
            },
            {
                "tag": "XCUIElementTypeButton",
                "attrs": {"enabled": "true", "visible": "true", "name": "取消"}
            }
        ]
        alertCancelSoup = CommonUtils.bsChainFind(soup, alertCancelChainList)
        if alertCancelSoup:
            self.clickElementCenterPosition(alertCancelSoup)
            foundAndProcessedPopup = True
        return foundAndProcessedPopup
    
```

iOSProcessKagsPopupStillNotLogin

```

def iOSProcessKagsPopupStillNotLogin(self, soup):
    foundAndProcessedPopup = False
    """
        康爱公社 弹框 您还未登录:
            <XCUIElementTypeOther type="XCUIElementTypeOther" name="pages/home1/home1[2]" label="pages/home1/home1[2]" enabled="true" visible="false" x="0" y="0" width="414" height="2707">
                . . .
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="135" y="373" width="144" height="30">
                        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="您还未登录" name="您还未登录" label="您还未登录" enabled="true" visible="true" x="146" y="373" width="122" height="30"/>
                    </XCUIElementTypeOther>
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="135" y="400" width="144" height="44">
                        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="请先登录后再进行操作" name="请先登录后再进行操作" label="请先登录后再进行操作" enabled="true" visible="true" x="135" y="420" width="144" height="19"/>
                    </XCUIElementTypeOther>
    
```

```

        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="112" y="473" width="70" height="22">
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="暂不登录" name="暂不登录" label="暂不登录" enabled="true" visible="true" x="112" y="473" width="70" height="22"/>
        </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="232" y="473" width="70" height="22">
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="立即登录" name="立即登录" label="立即登录" enabled="true" visible="true" x="232" y="473" width="70" height="22"/>
        </XCUIElementTypeOther>
    </XCUIElementTypeOther>
"""

stillNotLoginChainList = [
    {
        "tag": "XCUIElementTypeOther",
        "attrs": {"enabled": "true", "x": "0", "width": "%s" % self.X}
    },
    {
        "tag": "XCUIElementTypeOther",
        "attrs": {"visible": "true", "enabled": "true"}
    },
    {
        "tag": "XCUIElementTypeStaticText",
        "attrs": {"visible": "true", "enabled": "true", "value": "您还未登录"}
    },
]
stillNotLoginSoup = CommonUtils.bsChainFind(soup, stillNotLoginChainList)
if stillNotLoginSoup:
    parentOtherSoup = stillNotLoginSoup.parent
    parentParentOtherSoup = parentOtherSoup.parent
    if parentParentOtherSoup:
        tempNotLoginSoup = parentParentOtherSoup.find(
            "XCUIElementTypeStaticText",
            attrs={"visible": "true", "enabled": "true", "value": "暂不登录"})
    if tempNotLoginSoup:
        self.clickElementCenterPosition(tempNotLoginSoup)
        foundAndProcessedPopup = True
return foundAndProcessedPopup

```



iOSProcessWillOpenNonOfficialPage

```

def iOSProcessWillOpenNonOfficialPage(self, soup):
    foundAndProcessed = False

    """
        将要访问 非有赞官方网页, 请确认是否继续访问
        <XCUIElementTypeWebView type="XCUIElementTypeWebView" enabled="true" visible="true"
    
```

```

    " x="0" y="64" width="375" height="603">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
    " x="0" y="64" width="375" height="603">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="0" width="0" height="0">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="0" width="0" height="0">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="64" width="375" height="603">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="64" width="375" height="603">
                        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="150" y="234" width="75" height="24">
                            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="将要访问" name="将要访问" label="将要访问" enabled="true" visible="true" x="150" y="234" width="75" height="23"/>
                        
                        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="http://weixintx.haoduoke.cn/mobile/new_shop?appId=wx63d48daf039fe35c" name="http://weixintx.haoduoke.cn/mobile/new_shop?appId=wx63d48daf039fe35c" label="http://weixintx.haoduoke.cn/mobile/new_shop?appId=wx63d48daf039fe35c" enabled="true" visible="true" x="37" y="270" width="301" height="40"/>
                    
                        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="非有赞官方网页，请确认是否继续访问。" name="非有赞官方网页，请确认是否继续访问。" label="非有赞官方网页，请确认是否继续访问。" enabled="true" visible="true" x="58" y="324" width="259" height="18"/>
                    
                        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="有赞安全中心" name="有赞安全中心" label="有赞安全中心" enabled="true" visible="true" x="160" y="632" width="74" height="15"/>
                    

```

```

    {
        "tag": "XCUIElementTypeOther",
        "attrs": {"visible": "true", "enabled": "true", "x": "0", "width": "%s" % self.X}
    },
    {
        "tag": "XCUIElementTypeStaticText",
        "attrs": {"visible": "true", "enabled": "true", "value": "将要访问"}
    },
]
willOpenSoup = CommonUtils.bsChainFind(soup, willOpenChainList)
if willOpenSoup:
    parentOtherSoup = willOpenSoup.parent
    parentParentOtherSoup = parentOtherSoup.parent
    if parentParentOtherSoup:
        continueOpenSoup = parentParentOtherSoup.find(
            "XCUIElementTypeButton",
            attrs={"visible": "true", "enabled": "true", "name": "继续访问"})
    )
    if continueOpenSoup:
        self.clickElementCenterPosition(continueOpenSoup)
        foundAndProcessed = True

return foundAndProcessed

```

iOSProcessSettingsAllowUseLocation

```

def iOSProcessSettingsAllowUseLocation(self, soup):
    """处理 设置中的 使用我的地理位置，并返回前一页
    注：往往是出现 弹框-是否授权当前位置，点击 允许后，跳转到此处设置页面
    """
    """
    设置 使用我的地理位置：
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
        " x="0" y="0" width="375" height="667">
            <XCUIElementTypeNavigationBar type="XCUIElementTypeNavigationBar" name="设置"
            enabled="true" visible="true" x="0" y="20" width="375" height="44">
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="返回" label="返回"
                enabled="true" visible="true" x="10" y="20" width="30" height="44"/>
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="设置"
                " name="设置" label="设置" enabled="true" visible="true" x="187" y="24" width="1" height="3
                6"/>
            </XCUIElementTypeNavigationBar>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
            true" x="0" y="0" width="375" height="667">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visib
                le="true" x="0" y="0" width="375" height="667">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" v
                    isible="true" x="0" y="0" width="375" height="667">
                        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="tr
                        ue" visible="false" x="0" y="0" width="375" height="64"/>

```

```

        <XCUIElementTypeTable type="XCUIElementTypeTable" enabled="true"
e" visible="true" x="0" y="0" width="375" height="667">
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText"
" value="允许&quot;贝德玛会员中心&quot;" name="允许&quot;贝德玛会员中心&quot;" label="允许&quo
t;贝德玛会员中心&quot;" enabled="true" visible="true" x="15" y="88" width="345" height="17"/>
            <XCUIElementTypeCell type="XCUIElementTypeCell" value="0"
enabled="true" visible="true" x="0" y="115" width="375" height="44">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enab
led="true" visible="true" x="0" y="115" width="375" height="1"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enab
led="true" visible="true" x="0" y="158" width="375" height="1"/>
                <XCUIElementTypeStaticText type="XCUIElementTypeStatic
Text" value="使用我的地理位置" name="使用我的地理位置" label="使用我的地理位置" enabled="true"
visible="true" x="15" y="126" width="139" height="22"/>
                <XCUIElementTypeSwitch type="XCUIElementTypeSwitch" va
lue="0" name="使用我的地理位置" label="使用我的地理位置" enabled="true" visible="true" x="309
" y="122" width="51" height="31"/>
            </XCUIElementTypeCell>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled=
"true" visible="true" x="0" y="159" width="375" height="1219"/>
        </XCUIElementTypeTable>
    </XCUIElementTypeOther>
</XCUIElementTypeOther>
</XCUIElementTypeOther>
</XCUIElementTypeOther>
</XCUIElementTypeOther>
"""
foundAndProcessedPopup = False
useLocationSwitchQuery = {"type": "XCUIElementTypeSwitch", "name": "使用我的地理位置", "en
abled": "true"}
parentCellClassChain = "/XCUIElementTypeCell[`enabled = 1 AND rect.width = %d`]" % self
.x
useLocationSwitchQuery["parent_class_chains"] = [parentCellClassChain]
foundAndClickeUseLocation = self.findAndClickElement(useLocationSwitchQuery, isShowErr
Log False)
logging.debug("foundAndClickeUseLocation=%s", foundAndClickeUseLocation)
if foundAndClickeUseLocation:
    parentNaviBarClassChain = '/XCUIElementTypeNavigationBar[`enabled = 1 AND name="设
置"]'
    backButtonQuery = {"type": "XCUIElementTypeButton", "name": "返回", "en
abled": "true"}
    backButtonQuery["parent_class_chains"] = parentNaviBarClassChain
    foundAndClickeBack = self.findAndClickElement(backButtonQuery, isShowErrLog False)
    logging.debug("foundAndClickeBack=%s", foundAndClickeBack)
    foundAndProcessedPopup = foundAndClickeBack

return foundAndProcessedPopup

```

iOSProcessPopupNetworkNotStableRetry

```

def iOSProcessPopupNetworkNotStableRetry(self, soup):
    """Process iOS popup 京东金融 网络不稳定,请点击重试"""
    foundAndProcessed = False
    """
        京东金融 页面 网络不稳定 请点击重试:
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
    " x="0" y="0" width="414" height="736">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
    true" x="0" y="0" width="414" height="64">
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="com icon bla
    ck backup u" label="com icon black backup u" enabled="true" visible="true" x="0" y="20" wi
    dth="44" height="44"/>
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" enabled="t
    rue" visible="false" x="65" y="20" width="284" height="44"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="community top
    report@3x" label="community top report@3x" enabled="true" visible="true" x="326" y="20" wi
    dth="44" height="44"/>
            </XCUIElementTypeOther>
            <XCUIElementTypeTable type="XCUIElementTypeTable" enabled="true" visible="
    true" x="0" y="64" width="414" height="673">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visib
    le="true" x="0" y="64" width="414" height="1"/>
                <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible
    ="true" x="0" y="64" width="414" height="629">
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value=
    "网络不稳定,请点击重试" name="网络不稳定,请点击重试" label="网络不稳定,请点击重试" enabled="true
    " visible="true" x="16" y="322" width="382" height="21"/>
                    <XCUIElementTypeImage type="XCUIElementTypeImage" name="icon_netwo
    rk-instability" enabled="true" visible="false" x="132" y="152" width="150" height="151"/>
                </XCUIElementTypeCell>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visib
    le="true" x="0" y="692" width="414" height="1"/>
            </XCUIElementTypeTable>
        </XCUIElementTypeOther>
    """
    networkNotStableChainList = [
        {
            "tag": "XCUIElementTypeTable",
            "attrs": {"enabled": "true", "visible": "true", "x": "0", "width": "%s" % self.X}
        },
        {
            "tag": "XCUIElementTypeCell",
            "attrs": {"enabled": "true", "visible": "true", "x": "0", "width": "%s" % self.X}
        },
        {
            "tag": "XCUIElementTypeStaticText",
            "attrs": {"enabled": "true", "visible": "true", "value": "网络不稳定,请点击重试"}
        }
    ]
    networkNotStableSoup = CommonUtils.bsChainFind(soup, networkNotStableChainList)
    if networkNotStableSoup:
        self.clickElementCenterPosition(networkNotStableSoup)
        foundAndProcessed = True

```

```
    return foundAndProcessed
```

iOSProcessPopupNetworkNotStableRefresh

```
def iOSProcessPopupNetworkNotStableRefresh(self, soup):
    """Process iOS popup 京东金融 页面 网络不稳定 刷新试试"""
    foundAndProcessedPopup = False
    """
        京东金融 页面 网络不稳定 刷新试试:
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
    " x="0" y="64" width="414" height="623">
                <XCUIElementTypeTable type="XCUIElementTypeTable" name="空列表" label="空列
表" enabled="true" visible="false" x="0" y="64" width="414" height="623"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
true" x="0" y="64" width="414" height="623">
                    <XCUIElementTypeImage type="XCUIElementTypeImage" name="com_network_er
r" enabled="true" visible="false" x="152" y="164" width="110" height="110"/>
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="网络
不稳定" name="网络不稳定" label="网络不稳定" enabled="true" visible="true" x="0" y="287" widt
h="414" height="17"/>
                    <XCUIElementTypeButton type="XCUIElementTypeButton" name="刷新试试" lab
el="刷新试试" enabled="true" visible="true" x="147" y="323" width="120" height="41"/>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
    """
    tryRefreshChainList = [
        {
            "tag": "XCUIElementTypeOther",
            "attrs": {"enabled": "true", "visible": "true", "x": "0", "width": "%s" % self.X}
        },
        {
            "tag": "XCUIElementTypeOther",
            "attrs": {"enabled": "true", "visible": "true", "x": "0", "width": "%s" % self.X}
        },
        {
            "tag": "XCUIElementTypeButton",
            "attrs": {"enabled": "true", "visible": "true", "name": "刷新试试"}
        },
    ]
    tryRefreshSoup = CommonUtils.bsChainFind(soup, tryRefreshChainList)
    if tryRefreshSoup:
        self.clickElementCenterPosition(tryRefreshSoup)
        foundAndProcessedPopup = True

        # # Special: above click by position not work
        # # so change to wda query element then click
        # tryRefreshButtonQuery = {"type": "XCUIElementTypeButton", "enabled": "true", "name
": "刷新试试"}
        # foundAndClicked = findAndClickElement(curSession, tryRefreshButtonQuery)
        # foundAndProcessedPopup = foundAndClicked
```

```
    return foundAndProcessedPopup
```

iOSProcessPopupCertificateError

```
def iOSProcessPopupCertificateError(self, soup):
    """Process iOS popup 安全证书存在问题 网络出错, 轻触屏幕重新加载"""
    foundAndProcessed = False

    """
        <XCUIElementTypeButton type="XCUIElementTypeButton" name="网络出错, 轻触屏幕重新加载:-1202" label="网络出错, 轻触屏幕重新加载:-1202" enabled="true" visible="true" x="0" y="64" width="414" height="672">
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="网络出错, 轻触屏幕重新加载:-1202" name="网络出错, 轻触屏幕重新加载:-1202" label="网络出错, 轻触屏幕重新加载:-1202" enabled="true" visible="true" x="0" y="264" width="414" height="40"/>
        </XCUIElementTypeButton>
    """

    netErrTouchReloadP = re.compile("网络出错, 轻触屏幕重新加载")
    foundTouchReload = soup.find(
        "XCUIElementTypeButton",
        attrs={"visible": "true", "name": netErrTouchReloadP, "enabled": "true"}
    )
    if foundTouchReload:
        self.clickElementCenterPosition(foundTouchReload)
        foundAndProcessed = True

    if not foundAndProcessed:
        """
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="网络出错, 轻触屏幕重新加载:-1202" label="网络出错, 轻触屏幕重新加载:-1202" enabled="true" visible="false" x="0" y="64" width="414" height="672">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="网络出错, 轻触屏幕重新加载:-1202" name="网络出错, 轻触屏幕重新加载:-1202" label="网络出错, 轻触屏幕重新加载:-1202" enabled="true" visible="false" x="0" y="264" width="414" height="40"/>
            </XCUIElementTypeButton>

            <XCUIElementTypeWindow type="XCUIElementTypeWindow" enabled="true" visible="true" x="0" y="0" width="414" height="736">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="736">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="736">
                        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="736">
                            <XCUIElementTypeButton type="XCUIElementTypeButton" enabled="true" visible="true" x="0" y="0" width="414" height="736">
                                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="安全警告" name="安全警告" label="安全警告" enabled="true" visible="true" x="71" y="286" width="272" height="19"/>
                                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="mp.weixin.qq.com 该网站的安全证书存在问题, 可选择“继续”在浏览器中访问。" name="mp.weixin.qq.co
```

```

    m 该网站的安全证书存在问题，可选择“继续”在浏览器中访问。" label="mp.weixin.qq.com 该网站的安全证
    书存在问题，可选择“继续”在浏览器中访问。" enabled="true" visible="true" x="71" y="320" width=""
    272" height="74"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value=
    "取消" name="取消" label="取消" enabled="true" visible="true" x="109" y="443" width="36" hei
    ght="22"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value=
    "继续" name="继续" label="继续" enabled="true" visible="true" x="269" y="443" width="36" hei
    ght="22"/>
            </XCUIElementTypeButton>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    </XCUIElementTypeWindow>
    """
    securityWarningChainList = self.generateCommonPopupItemList(thirdLevelValue =
    "安全警告")
    securityWarningSoup = CommonUtils.bsChainFind(soup, securityWarningChainList)
    if securityWarningSoup:
        parentButtonSoup = securityWarningSoup.parent
        if parentButtonSoup:
            certificateProblemP = re.compile("该网站的安全证书存在问题")
            foundCertProblem = parentButtonSoup.find(
                "XCUIElementTypeStaticText",
                attrs={"visible": "true", "enabled": "true", "value": certificateProblemP})
            )
            if foundCertProblem:
                foundCancel = parentButtonSoup.find(
                    "XCUIElementTypeStaticText",
                    attrs={"visible": "true", "enabled": "true", "value": "取消"})
            )
            if foundCancel:
                self.clickElementCenterPosition(foundCancel)
                foundAndProcessed = True
    return foundAndProcessed

```

iOSProcessPopupWeixinLogin

```

def iOSProcessPopupWeixinLogin(self, soup):
    """Process iOS popup 微信登录"""
    foundAndProcessedPopup = False
    """
        <XCUIElementTypeWindow type="XCUIElementTypeWindow" enabled="true" visible="true"
    x="0" y="0" width="414" height="736">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true
    " x="0" y="0" width="414" height="736">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true

```

```

    " x="0" y="0" width="414" height="736">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="736">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="736">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="52" y="236" width="310" height="263">
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="微信登录" name="微信登录" label="微信登录" enabled="true" visible="true" x="83" y="251" width="248" height="18"/>
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="52" y="285" width="310" height="1"/>
                        <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="false" x="189" y="306" width="36" height="35"/>
                            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="微店消息助手申请获得以下权限:" name="微店消息助手申请获得以下权限:" label="微店消息助手申请获得以下权限:" enabled="true" visible="true" x="82" y="356" width="231" height="19"/>
                            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="82" y="390" width="250" height="1"/>
                                <XCUIElementTypeTable type="XCUIElementTypeTable" enabled="true" visible="true" x="82" y="391" width="250" height="62">
                                    <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" visible="true" x="82" y="391" width="250" height="47">
                                        <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="false" x="82" y="406" width="15" height="14"/>
                                            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="获得你的公开信息（昵称、头像、地区及性别）" name="获得你的公开信息（昵称、头像、地区及性别）" label="获得你的公开信息（昵称、头像、地区及性别）" enabled="true" visible="true" x="101" y="405" width="230" height="34"/>
                                    
                                
                            
                            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="52" y="454" width="310" height="1"/>
                                <XCUIElementTypeButton type="XCUIElementTypeButton" name="拒绝" label="拒绝" enabled="true" visible="true" x="52" y="454" width="155" height="45">
                                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="拒绝" name="拒绝" label="拒绝" enabled="true" visible="true" x="112" y="466" width="35" height="21"/>
                                
                            
                            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="207" y="454" width="1" height="45"/>
                                <XCUIElementTypeButton type="XCUIElementTypeButton" name="允许" label="允许" enabled="true" visible="true" x="207" y="454" width="155" height="45">
                                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="允许" name="允许" label="允许" enabled="true" visible="true" x="267" y="466" width="35" height="21"/>
                                
                            
                        
                    
                
            
        
    

```

```

"""
weixinLoginChainList = self.generateCommonPopupItemChainList(
    secondLevelTag="XCUIElementTypeOther",
    thirdLevelValue="微信登录"
)
foundWeixinLogin = CommonUtils.bsChainFind(soup, weixinLoginChainList)

tableChainList = self.generateCommonPopupItemChainList(
    secondLevelTag="XCUIElementTypeOther",
    thirdLevelTag="XCUIElementTypeTable",
)
foundTable = CommonUtils.bsChainFind(soup, tableChainList)

if foundWeixinLogin and foundTable:
    parentOtherSoup = foundWeixinLogin.parent
    if parentOtherSoup:
        foundAllowButton = parentOtherSoup.find(
            "XCUIElementTypeButton",
            attrs={"visible": "true", "enabled": "true", "label": "允许"}
        )
        if foundAllowButton:
            self.clickElementCenterPosition(foundAllowButton)
            foundAndProcessedPopup = True

return foundAndProcessedPopup

```

iOSProcessPopupJumpThirdApp

```

def iOSProcessPopupJumpThirdApp(self, soup):
    """Process iOS popup 可能离开微信，打开第三方应用"""
    foundAndProcessedPopup = False

    """
        <XCUIElementTypeWindow type="XCUIElementTypeWindow" enabled="true" visible="true"
        x="0" y="0" width="414" height="736">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
            " x="0" y="0" width="414" height="736">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
                true" x="0" y="0" width="414" height="736">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visib
                    le="true" x="0" y="0" width="414" height="736">
                        <XCUIElementTypeButton type="XCUIElementTypeButton" enabled="true"
                        visible="true" x="0" y="0" width="414" height="736">
                            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" en
                            abled="true" visible="false" x="47" y="288" width="0" height="0"/>
                            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" va
                            lue="可能离开微信，打开第三方应用" name="可能离开微信，打开第三
                            方应用" enabled="true" visible="true" x="71" y="331" width="272"
                            height="18"/>
                            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" va
                            lue="取消" name="取消" label="取消" enabled="true" visible="true" x="109"
                            y="409" width="36"
    
```

```

height="22"/>
<XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="继续" name="继续" label="继续" enabled="true" visible="true" x="269" y="409" width="36" height="22"/>
</XCUIElementTypeButton>
</XCUIElementTypeOther>
</XCUIElementTypeOther>
</XCUIElementTypeOther>
</XCUIElementTypeWindow>
"""

leaveWeixinChainList = self.generateCommonPopupItemChainList(thirdLevelValue "可能离开
微信, 打开第三方应用")
foundLeaveWeixin = CommonUtils.bsChainFind(soup, leaveWeixinChainList)
if foundLeaveWeixin:
    parentButtonSoup = foundLeaveWeixin.parent
    if parentButtonSoup:
        foundCancel = parentButtonSoup.find(
            "XCUIElementTypeStaticText",
            attrs={"visible": "true", "enabled": "true", "value": "取消"})
    )
    if foundCancel:
        self.clickElementCenterPosition(foundCancel)
        foundAndProcessedPopup = True

return foundAndProcessedPopup

```

iOSProcessPopupDoSomeFail

```

# def iOSProcessPopupGetInfoFail(self, soup):
def iOSProcessPopupDoSomeFail(self, soup):
    """Process iOS popup 获取信息失败 微信登录失败 等"""
    foundAndProcessedPopup = False
    """
    微信:
        获取信息失败:
            <XCUIElementTypeWindow type="XCUIElementTypeWindow" enabled="true" visible="tr
ue" x="0" y="0" width="414" height="736">
                ...
                    <XCUIElementTypeButton type="XCUIElementTypeButton" enabled="true" visible
="true" x="0" y="0" width="414" height="736">
                        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="获取
信息失败" name="获取信息失败" label="获取信息失败" enabled="true" visible="true" x="71" y="30
9" width="272" height="21"/>
                            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="取消
" name="取消" label="取消" enabled="true" visible="true" x="109" y="419" width="36" height=
"22"/>
                        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="确定
" name="确定" label="确定" enabled="true" visible="true" x="269" y="419" width="36" height=
"22"/>
                    </XCUIElementTypeButton>

```

```

        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="320" y="24" width="87" height="32">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="320" y="24" width="88" height="32">
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="更多" label="更多" enabled="true" visible="true" x="320" y="24" width="44" height="32"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="关闭" label="关闭" enabled="true" visible="true" x="363" y="24" width="45" height="32"/>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
        . . .
    </XCUIElementTypeWindow>

```

微信登录失败:

```

    <XCUIElementTypeWindow type="XCUIElementTypeWindow" enabled="true" visible="true" x="0" y="0" width="375" height="667">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="375" height="667">
            <XCUIElementTypeButton type="XCUIElementTypeButton" enabled="true" visible="true" x="0" y="0" width="375" height="667"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="32" y="247" width="311" height="173">
                <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="false" x="32" y="247" width="311" height="173"/>
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="微信登录失败" name="微信登录失败" label="微信登录失败" enabled="true" visible="true" x="52" y="272" width="271" height="22"/>
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="此公众号并没有这些scope的权限，错误码:10005" name="此公众号并没有这些scope的权限，错误码:10005" label="此公众号并没有这些scope的权限，错误码:10005" enabled="true" visible="true" x="59" y="309" width="257" height="40"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="确定" label="确定" enabled="true" visible="true" x="32" y="370" width="311" height="50"/>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    </XCUIElementTypeWindow>

```

小程序:

```

    获取信息失败:
    <XCUIElementTypeButton type="XCUIElementTypeButton" enabled="true" visible="true" x="0" y="0" width="375" height="667">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="32" y="251" width="311" height="165">
            <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="false" x="32" y="251" width="311" height="165"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="获取信息失败" name="获取信息失败" label="获取信息失败" enabled="true" visible="true" x="52" y="276" width="271" height="21"/>
            <XCUIElementTypeTextView type="XCUIElementTypeTextView" value="是否重新加载页面" enabled="true" visible="true" x="121" y="311" width="133" height="35"/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="取消" label="取消" enabled="true" visible="true" x="32" y="365" width="156" height="51"/>
        </XCUIElementTypeOther>
    </XCUIElementTypeButton>

```

```

        <XCUIElementTypeButton type="XCUIElementTypeButton" name="确定" label="确定" enabled="true" visible="true" x="187" y="365" width="156" height="51"/>
    </XCUIElementTypeOther>
</XCUIElementTypeButton>
"""

# getInfoFailChainList = self.generateCommonPopupItemList(thirdLevelValue="获取信息失败")
# getInfoFailSoup = CommonUtils.bsChainFind(soup, getInfoFailChainList)
# if getInfoFailSoup:
#     parentButtonOrOtherSoup = getInfoFailSoup.parent
doSomeFailP = re.compile("\S+失败$") # 获取信息失败，微信登录失败
doSomeFailChainList = self.generateCommonPopupItemList(thirdLevelValue=doSomeFailP)

doSomeFailSoup = CommonUtils.bsChainFind(soup, doSomeFailChainList)
if doSomeFailSoup:
    parentButtonOrOtherSoup = doSomeFailSoup.parent
    if parentButtonOrOtherSoup:
        # 微信 弹框：获取信息失败
        foundConfirm = parentButtonOrOtherSoup.find(
            "XCUIElementTypeStaticText",
            attrs={"visible": "true", "enabled": "true", "value": "确定"})
    if not foundConfirm:
        # 小程序 弹框：获取信息失败
        # 微信 弹框：微信登录失败
        foundConfirm = parentButtonOrOtherSoup.find(
            "XCUIElementTypeButton",
            attrs={"visible": "true", "enabled": "true", "name": "确定"})
    if foundConfirm:
        self.clickElementCenterPosition(foundConfirm)
        foundAndProcessedPopup = True

return foundAndProcessedPopup

```

iOSProcessPopupWeixinAuthorizeLocation

```

def iOSProcessPopupWeixinAuthorizeLocation(self, soup):
    """Process iOS popup 是否授权当前位置"""
    foundAndProcessedPopup = False
    """

    微信 弹框 是否授权当前位置：
        <XCUIElementTypeButton type="XCUIElementTypeButton" enabled="true" visible="true" x="0" y="0" width="375" height="667">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="32" y="240" width="311" height="187">
                <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="false" x="32" y="240" width="311" height="187"/>

```

```

        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="是否授权当前位置" name="是否授权当前位置" label="是否授权当前位置" enabled="true" visible="true" x="52" y="265" width="271" height="21"/>
        <XCUIElementTypeTextView type="XCUIElementTypeTextView" value="需要获取您的地理位置,请确认授权,否则无法获取专相导航" enabled="true" visible="true" x="67" y="300" width="241" height="57"/>
        <XCUIElementTypeButton type="XCUIElementTypeButton" name="取消" label="取消" enabled="true" visible="true" x="32" y="376" width="156" height="51"/>
        <XCUIElementTypeButton type="XCUIElementTypeButton" name="确定" label="确定" enabled="true" visible="true" x="187" y="376" width="156" height="51"/>
    </XCUIElementTypeOther>
</XCUIElementTypeButton>
"""

authorizeLocationChainList = [
{
    "tag": "XCUIElementTypeButton",
    "attrs": {"visible": "true", "enabled": "true", "width": "%s" % self.X, "height": "%s" % self.totally}
},
{
    "tag": "XCUIElementTypeOther",
    "attrs": {"visible": "true", "enabled": "true"}
},
{
    "tag": "XCUIElementTypeStaticText",
    "attrs": {"visible": "true", "enabled": "true", "value": "是否授权当前位置"}
},
]
authorizeLocationSoup = CommonUtils.bsChainFind(soup, authorizeLocationChainList)
if authorizeLocationSoup:
    parentOtherSoup = authorizeLocationSoup.parent
    confirmSoup = parentOtherSoup.find(
        "XCUIElementTypeButton",
        attrs={"visible": "true", "enabled": "true", "name": "确定"})
)
if confirmSoup:
    self.clickElementCenterPosition(confirmSoup)
    foundAndProcessedPopup = True

return foundAndProcessedPopup

```

iOSProcessPopupMiniprogramWarning

```

def iOSProcessPopupMiniprogramWarning(self, soup):
"""
    微信-小程序 弹框 警告 尚未进行授权,请点击确定跳转到授权页面进行授权
"""
foundAndProcessedPopup = False
"""
    微信-小程序 弹框 警告 尚未进行授权:

```

```

        <XCUIElementTypeButton type="XCUIElementTypeButton" enabled="true" visible="true" x="0" y="0" width="375" height="667">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="32" y="240" width="311" height="187">
                <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="false" x="32" y="240" width="311" height="187"/>
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="警告" name="警告" label="警告" enabled="true" visible="true" x="52" y="265" width="271" height="21"/>
                <XCUIElementTypeTextView type="XCUIElementTypeTextView" value="尚未进行授权, 请点击确定跳转到授权页面进行授权。" enabled="true" visible="true" x="60" y="300" width="255" height="57"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="取消" label="取消" enabled="true" visible="true" x="32" y="376" width="156" height="51"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton" name="确定" label="确定" enabled="true" visible="true" x="187" y="376" width="156" height="51"/>
            </XCUIElementTypeOther>
        </XCUIElementTypeButton>
    """
warningChainList = [
    {
        "tag": "XCUIElementTypeButton",
        "attrs": {"visible": "true", "enabled": "true", "width": "%s" % self.X, "height": "%s" % self.totally}
    },
    {
        "tag": "XCUIElementTypeOther",
        "attrs": {"visible": "true", "enabled": "true"}
    },
    {
        "tag": "XCUIElementTypeStaticText",
        "attrs": {"visible": "true", "enabled": "true", "value": "警告"}
    }
]
warningSoup = CommonUtils.bsChainFind(soup, warningChainList)
if warningSoup:
    parentOtherSoup = warningSoup.parent
    confirmSoup = parentOtherSoup.find(
        "XCUIElementTypeButton",
        attrs={"visible": "true", "enabled": "true", "name": "确定"})
)
if confirmSoup:
    self.clickElementCenterPosition(confirmSoup)
    foundAndProcessedPopup = True

return foundAndProcessedPopup

```

iOSProcessPopupMiniprogramGetYour

```
# def iOSProcessPopupMiniprogramAuthority(self, soup):
```

```

def iOSProcessPopupMiniprogramGetYour(self, soup):
    """Process iOS popup 获取你的昵称、头像、地区及性别 / 获取你的通讯地址 / 需要获取你的地理位置"""
    foundAndProcessedPopup = False
    """
        弹框 获取你的昵称、头像、地区及性别:
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
    " x="0" y="0" width="375" height="667">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
    " x="0" y="0" width="375" height="667"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
    " x="0" y="315" width="375" height="352">
                <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="false"
    x="20" y="335" width="24" height="25"/>
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="优理
    氏UNES官方旗舰店" name="优理氏UNES官方旗舰店" label="优理氏UNES官方旗舰店" enabled="true" visi
    ble="true" x="49" y="338" width="163" height="19"/>
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="申请
    " name="申请" label="申请" enabled="true" visible="true" x="217" y="338" width="31" height=
    "19"/>
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="获取
    你的昵称、头像、地区及性别" name="获取你的昵称、头像、地区及性别" label="获取你的昵称、头像、地区
    及性别" enabled="true" visible="true" x="20" y="383" width="260" height="25"/>
                <XCUIElementTypeTable type="XCUIElementTypeTable" enabled="true" visib
    le="true" x="20" y="423" width="335" height="65">
                    <XCUIElementTypeCell type="XCUIElementTypeCell" enabled="true" vis
    ible="true" x="20" y="423" width="335" height="65">
                        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="tru
    e" visible="true" x="20" y="423" width="335" height="65"/>
                        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="tru
    e" visible="false" x="20" y="423" width="335" height="1"/>
                        <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="tru
    e" visible="false" x="20" y="435" width="40" height="41"/>
                        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="tru
    e" visible="false" x="20" y="486" width="335" height="1"/>
                        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" va
    lue="斐波测试2" name="斐波测试2" label="斐波测试2" enabled="true" visible="false" x="72" y="4
    35" width="291" height="22"/>
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" va
    lue="微信个人信息" name="微信个人信息" label="微信个人信息" enabled="true" visible="false" x=
    "72" y="460" width="291" height="18"/>
                    <XCUIElementTypeImage type="XCUIElementTypeImage" name="icon_s
    elected" enabled="true" visible="false" x="335" y="445" width="20" height="21"/>
                </XCUIElementTypeCell>
            </XCUIElementTypeTable>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="使用其他头像和
    昵称" label="使用其他头像和昵称" enabled="true" visible="true" x="20" y="503" width="138" he
    ight="18"/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="取消" label="取
    消" enabled="true" visible="true" x="59" y="561" width="121" height="40"/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="允许" label="允
    许" enabled="true" visible="true" x="195" y="561" width="121" height="40"/>
        </XCUIElementTypeOther>
    
```

```
</XCUIElementTypeOther>
```

弹框 获取你的通讯地址:

```

<XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
" x="0" y="0" width="375" height="667">
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
x="0" y="0" width="375" height="667"/>
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
x="0" y="429" width="375" height="238">
        <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="false"
x="20" y="449" width="24" height="25"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="优理
氏UNES官方旗舰店" name="优理氏UNES官方旗舰店" label="优理氏UNES官方旗舰店" enabled="true" visi
ble="true" x="49" y="452" width="163" height="19"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="申请
" name="申请" label="申请" enabled="true" visible="true" x="217" y="452" width="31" height=
"19"/>
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="获取
你的通讯地址" name="获取你的通讯地址" label="获取你的通讯地址" enabled="true" visible="true" x
="20" y="497" width="139" height="25"/>
        <XCUIElementTypeButton type="XCUIElementTypeButton" name="拒绝" label="拒绝"
enabled="true" visible="true" x="59" y="561" width="121" height="40"/>
        <XCUIElementTypeButton type="XCUIElementTypeButton" name="允许" label="允许"
enabled="true" visible="true" x="195" y="561" width="121" height="40"/>
    </XCUIElementTypeOther>
</XCUIElementTypeOther>
```

小程序 弹框 需要获取你的地理位置:

```

<XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
" x="0" y="0" width="375" height="667">
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
false" x="0" y="0" width="375" height="667"/>
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="
true" x="0" y="0" width="375" height="667">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visib
le="true" x="0" y="0" width="375" height="667"/>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visib
le="false" x="0" y="0" width="375" height="667"/>
        <XCUIElementTypeAlert type="XCUIElementTypeAlert" name="贝德玛会
员中心" label="贝德玛会员中心" enabled="true" visible="true" x="52" y="270" width="271" height="127">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" v
isible="true" x="52" y="270" width="271" height="127">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="tr
ue" visible="true" x="52" y="270" width="271" height="127">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled=
"true" visible="false" x="52" y="270" width="271" height="127">
                        <XCUIElementTypeOther type="XCUIElementTypeOther" enab
led="true" visible="false" x="52" y="270" width="271" height="127"/>
                        <XCUIElementTypeOther type="XCUIElementTypeOther" enab
led="true" visible="false" x="52" y="270" width="271" height="127"/>
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled=
"true" visible="false" x="52" y="270" width="271" height="127"/>
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true"
visible="false" x="52" y="270" width="271" height="127"/>
```

```

                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="52" y="270" width="271" height="127"/>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="52" y="270" width="271" height="127">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="52" y="270" width="271" height="83">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="贝德玛会员中心" name="贝德玛会员中心" label="贝德玛会员中心" need="需要获取你的地理位置" enabled="true" visible="true" x="68" y="290" width="239" height="43"/>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="52" y="352" width="271" height="1">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="52" y="352" width="271" height="1"/>
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="52" y="352" width="271" height="1"/>
        </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="52" y="353" width="271" height="44">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="52" y="353" width="271" height="44">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="52" y="353" width="271" height="44">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="52" y="353" width="136" height="44">
                        <XCUIElementTypeButton type="XCUIElementTypeButton" name="不允许" label="不允许" enabled="true" visible="true" x="52" y="353" width="136" height="44"/>
                    </XCUIElementTypeOther>
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="187" y="353" width="1" height="44">
                        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="187" y="353" width="1" height="44"/>
                        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="187" y="353" width="1" height="44"/>
                    </XCUIElementTypeOther>
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="188" y="353" width="135" height="44">
                        <XCUIElementTypeButton type="XCUIElementTypeButton" name="允许" label="允许" enabled="true" visible="true" x="188" y="353" width="135" height="44"/>
                    </XCUIElementTypeOther>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    </XCUIElementTypeOther>
</XCUIElementTypeOther>

```

```

                </XCUIElementTypeOther>
            </XCUIElementTypeAlert>
        </XCUIElementTypeOther>
    </XCUIElementTypeOther>
"""

# getYourP = re.compile("获取你的\S+") # 获取你的通讯地址， 获取你的昵称、头像、地区及性别
getYourP = re.compile("(需要)?获取你的\S+") # 获取你的通讯地址， 获取你的昵称、头像、地区及性别，需要获取你的地理位置
getYourChainList = [
    {
        "tag": "XCUIElementTypeOther",
        "attrs": {"visible": "true", "enabled": "true", "x": "0", "width": self.X}
    },
    {
        "tag": "XCUIElementTypeOther",
        "attrs": {"visible": "true", "enabled": "true", "x": "0", "width": self.X}
    },
    {
        "tag": "XCUIElementTypeStaticText",
        "attrs": {"visible": "true", "enabled": "true", "value": getYourP}
    },
]
getYourSoup = CommonUtils.bsChainFind(soup, getYourChainList)
if getYourSoup:
    parentOther = getYourSoup.parent
    parentParentOther = None
    parentParentParentOther = None

    allowTag = "XCUIElementTypeButton"
    allowAttrs = {"visible": "true", "enabled": "true", "name": "允许"}

    # 获取你的通讯地址，获取你的昵称、头像、地区及性别
    foundAllow = parentOther.find(allowTag, attrs=allowAttrs)

    if not foundAllow:
        # for support future possible case
        parentParentOther = parentOther.parent
        foundAllow = parentParentOther.find(allowTag, attrs=allowAttrs)

    if not foundAllow:
        # "贝德玛会员中心" 需要获取你的地理位置
        parentParentParentOther = parentParentOther.parent
        foundAllow = parentParentParentOther.find(allowTag, attrs=allowAttrs)

    if foundAllow:
        self.clickElementCenterPosition(foundAllow)
        foundAndProcessedPopup = True

return foundAndProcessedPopup

```

iOSProcessPopupMiniprogramDisclaimer

```

def iOSProcessPopupMiniprogramDisclaimer(self, soup):
    """Process iOS popup 免责声明"""
    foundAndProcessedPopup = False
    """
        <XCUIElementTypeWindow type="XCUIElementTypeWindow" enabled="true" visible="true"
    x="0" y="0" width="375" height="667">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true"
    " x="0" y="0" width="375" height="667">
                <XCUIElementTypeButton type="XCUIElementTypeButton" enabled="true" visible
    ="true" x="0" y="0" width="375" height="667">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visib
    le="true" x="27" y="208" width="321" height="251">
                        <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" v
    isible="false" x="27" y="208" width="321" height="251"/>
                        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value=
    "免责声明" name="免责声明" label="免责声明" enabled="true" visible="true" x="51" y="240" widt
    h="273" height="21"/>
                        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value=
    "本服务由第三方提供，获取你的微信收货地址信息。相关服务和责任由该第三方承担，如有问题请咨询该公司客
    服。" name="本服务由第三方提供，获取你的微信收货地址信息。相关服务和责任由该第三方承担，如有问题请
    咨询该公司客服。" label="本服务由第三方提供，获取你的微信收货地址信息。相关服务和责任由该第三方承担
    ，如有问题请咨询该公司客服。" enabled="true" visible="true" x="51" y="277" width="273" height
    ="94"/>
                        <XCUIElementTypeButton type="XCUIElementTypeButton" name="知道了" l
    abel="知道了" enabled="true" visible="true" x="27" y="403" width="321" height="56"/>
                    </XCUIElementTypeOther>
                </XCUIElementTypeButton>
            </XCUIElementTypeOther>
        </XCUIElementTypeWindow>
    """
    disclaimerChainList = [
        {
            "tag": "XCUIElementTypeWindow",
            "attrs": {"visible": "true", "enabled": "true", "width": self.X, "height": self.
totalY}
        },
        {
            "tag": "XCUIElementTypeButton",
            "attrs": {"visible": "true", "enabled": "true", "width": self.X, "height": self.
totalY}
        },
        {
            "tag": "XCUIElementTypeStaticText",
            "attrs": {"visible": "true", "enabled": "true", "value": "免责声明"}
        },
    ]
    disclaimerSoup = CommonUtils.bsChainFind(soup, disclaimerChainList)
    if disclaimerSoup:
        parentOtherSoup = disclaimerSoup.parent
        if parentOtherSoup:
            foundIKnow = parentOtherSoup.find(
                "XCUIElementTypeButton",
                attrs={"visible": "true", "enabled": "true", "name": "知道了"})

```

```

        )
    if foundIKnow:
        self.clickElementCenterPosition(foundIKnow)
        foundAndProcessedPopup = True

    return foundAndProcessedPopup

```

期间部分相关函数：

```

def generateCommonPopupItemChainList(self,
    firstLevelTag="XCUIElementTypeWindow",
    secondLevelTag="XCUIElementTypeButton",
    thirdLevelTag="XCUIElementTypeStaticText",
    thirdLevelValue=None,
    thirdLevelName=None,
):
    CommonAttrs_VisibleEnabledFullWidthFullHeight = {"visible": "true", "enabled": "true", "width": "%s" % self.X, "height": "%s" % self.totalY}
    commonItemChainList = [
        {
            "tag": firstLevelTag,
            "attrs": CommonAttrs_VisibleEnabledFullWidthFullHeight
        },
        {
            "tag": secondLevelTag,
            "attrs": CommonAttrs_VisibleEnabledFullWidthFullHeight
        },
    ]
    thirdItemAttrs = {"visible": "true", "enabled": "true"}
    if thirdLevelValue:
        thirdItemAttrs["value"] = thirdLevelValue
    if thirdLevelName:
        thirdItemAttrs["name"] = thirdLevelName
    thirdItemDict = {
        "tag": thirdLevelTag,
        "attrs": thirdItemAttrs
    }
    commonItemChainList.append(thirdItemDict)
    return commonItemChainList

```

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2020-08-09 10:13:38

调试辅助

在开发期间，往往要调试一些内容，比如页面元素有哪些，位置大小等如何，所以会想要一些函数，可以给图片上元素加框显示等等，这类函数，属于方便调试，辅助调试方面的内容。

之前实现了很多复制调试的函数，如下，供参考。

debugDrawElementRect debugDrawScreenRect 给图片画框

```

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

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

    return curImg

def debugDrawElementRect(self, elementList, curImgPath None, isShowEach=False, isSaveEach=True, isDrawInSinglePic=False):
    """for debug, to draw rectangle for each element in current screenshot"""
    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], curWidth, curHeight]
        curTimeStr = CommonUtils.getCurDatetimeStr("%H%M%S")
        curSaveTal = "_rect_{}_{}|{}|{}|{}|{}".format(curTimeStr) # '_rect_155618_{}|{}|{}|{}|{}'

        curInputImg = None
        if isDrawInSinglePic:
            curInputImg = curImg
        else:
            curInputImg = curImgPath
        curImg = CommonUtils.imageDrawRectangle(
            curInputImg,
            curRect,
    )

```

```

        isShow isShowEach,
        isAutoSave isSaveEach,
        saveTail curSaveTail,
        isDrawClickedPosCircle False,
    )

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

    return

```

调用：

```

# # for debug
# self.debugDrawScreenRect(curRect=bounds)

```

和：

```

Elements = self.get_elements_valuable(Elements)
# for debug
# self.debugLineElementRect(Elements, isShowEach=True, isSaveEach=False, isDrawInSinglePic
# =True)
self.debugLineElementRect(Elements, isShowEach False, isSaveEach False, isDrawInSinglePic
# True)

Elements = [element for element in Elements if self.is_element_usable(element, filterFirstP
age)]
# for debug
# self.debugLineElementRect(Elements, isShowEach=True, isSaveEach=False, isDrawInSinglePic
# =True)
self.debugLineElementRect(Elements, isShowEach False, isSaveEach False, isDrawInSinglePic
# True)

```

用于批量给多个元素加上框，输出到单个图片中 -> 方便调试时，知道哪些元素被当前一轮循环过滤掉了

scaleToOrginSize 缩放图片到原始大小

背景：

iPhone的scale往往都是2或3等值

用iOS的截图都是大图，希望缩写到原图

```

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

```

调用举例：

```

curScale = 3
if isResizeToOriginal:
    # and resize to original screen size
    savedImgFile = self.scaleToOrginSize(savedImgFile, curScale)

```

saveiOSScreenshot 保存当前屏幕截图（图片文件）

```

def saveiOSScreenshot(self, filePrefix "", imageFormat="jpg", saveFolder ""):
    """
        do screehsot for ios device and saved to jpg
        same screenshot image file size compare:
            png: 734KB
            jpg: 100KB
            so better to use jpg
    """
    savedScreenFile = None

    curDatetimeStr = CommonUtils.getCurDatetimeStr()
    # screenFilename = "%s_screen.%s" % (curDatetimeStr, imageFormat)
    screenFilename = "%s.%s" % (curDatetimeStr, imageFormat)
    if filePrefix:
        screenFilename = "%s_%s" % (filePrefix, screenFilename)
        # 'com.netease.cloudmusic_20200221_170305.jpg'
    screenFilename = os.path.join(saveFolder, screenFilename)

    if imageFormat == "png":
        curPillowObj = self.wdaClient.screenshot(png_filename screenFilename)
        savedScreenFile = screenFilename
    elif (imageFormat == "jpg") or (imageFormat == "jpeg"):
        curPillowObj = self.wdaClient.screenshot()
        # logging.debug("curPillowObj=%s", curPillowObj)
        # curPillowObj=<PIL.PngImagePlugin.PngImageFile image mode=RGB size=750x1334 at 0x
        10F6CEE80>
        # convert to PIL.Image and then save as jpg
        curPillowObj.save(screenFilename)

```

```

        savedScreenFile = screenFilename
    else:
        logging.debug("Unsupported image format: %s", imageFormat)

    if savedScreenFile:
        logging.debug("saved screenshot: %s", savedScreenFile)
    return savedScreenFile

```

调用举例 + 相关函数：

```

def debugiOSSaveScreenshot(self, saveFolder, isResizeToOriginal=True, curScale=2.0):
    """for debug, save iOS screenshot"""
    # # for if enable debug screenshot image content too long
    # # so disable debug screenshot
    # needEnableLater = False
    # if wda.DEBUG:
    #     wda.DEBUG = False
    #     needEnableLater = True
    # savedImgFile = self.saveIOSScreenshot(filePrefix="", saveFolder=saveFolder)
    # if needEnableLater:
    #     wda.DEBUG = True
    def _saveScreenshot():
        return self.saveIOSScreenshot(filePrefix="", saveFolder=saveFolder)
    savedImgFile = self.runButNotOutputWdaDebug(_saveScreenshot)

    logging.debug("Saved iOS screenshot %s", savedImgFile)
    if isResizeToOriginal:
        # and resize to original screen size
        savedImgFile = self.scaleToOrginSize(savedImgFile, curScale)
    return savedImgFile

```

调用：

```

    elif self.isiOS:
        fullImgFilePath = self.debugiOSSaveScreenshot(saveFolder=saveFolder, curScale=
self.curSession.scale)

```

以及：

```

def runButNotOutputWdaDebug(self, funcToRun):
    # disable then re-enable debug for
    # avoid debug print too many binary image data of current screenshot
    # for internal session scale will trigger do screenshot
    needEnableLater = False
    if wda.DEBUG:
        wda.DEBUG = False
        needEnableLater = True

```

```
returnValue = funcToRun()

if needEnableLater:
    wda.DEBUG = True

return returnValue
```

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09 10:13:38

源码分析

下面给出 `facebook-wda` 的部分源码的内部实现逻辑分析：

wda内部用到了Appium

文件： `refer/WebDriverAgent/WebDriverAgentLib/Commands/FBSessionCommands.m`

```
+ (NSArray *)routes
{
    return @{
        [[FBRoute POST @"/url"] respondWithTarget self action @selector(handleOpenURL:)],
        [[FBRoute POST @"/session"] withoutSession respondWithTarget self action @selector(handleCreateSession)],
        [[FBRoute POST @"/wda/apps/launch"] respondWithTarget self action @selector(handleSessionAppLaunch)],
        [[FBRoute POST @"/wda/apps/activate"] respondWithTarget self action @selector(handleSessionAppActivate)],
        [[FBRoute POST @"/wda/apps/terminate"] respondWithTarget self action @selector(handleSessionAppTerminate)],
        [[FBRoute POST @"/wda/apps/state"] respondWithTarget self action @selector(handleSessionAppState)],
        [[FBRoute GET @"/wda/apps/list"] respondWithTarget self action @selector(handleGetActiveAppsList)],
        [[FBRoute GET @""] respondWithTarget self action @selector(handleGetActiveSession)],
        [[FBRoute DELETE @""] respondWithTarget self action @selector(handleDeleteSession)],
        [[FBRoute GET @"/status"].withoutSession respondWithTarget self action @selector(handleGetStatus)],
        // Health check might modify simulator state so it should only be called in-between testing sessions
        [[FBRoute GET @"/wda/healthcheck"].withoutSession respondWithTarget self action @selector(handleGetHealthCheck)],
        // Settings endpoints
        [[FBRoute GET @"/appium/settings"] respondWithTarget self action @selector(handleGetSettings)],
        [[FBRoute POST @"/appium/settings"] respondWithTarget self action @selector(handleSetSettings)],
    };
}
```

中的 `/appium/settings` 就是Appium的。

对应着支持外部的python的wda去调用：

文件： `/Users/limao/.pyenv/versions/3.8.0/Python.framework/Versions/3.8/lib/python3.8/site-packages/wda/__init__.py`

```

def get_settings(self):
    resp = self.http.get("/appium/settings")
    respSettings = resp.value
    if DEBUG:
        # print("resp=%s" % resp) #  TypeError not all arguments converted during string f
        # ormatting
        print("respSettings=%s" % respSettings)
    return respSettings

def set_settings(self, newSettings):
    postResp = self.http.post("/appium/settings", {
        "settings": newSettings,
    })
    respNewSettings = postResp.value
    if DEBUG:
        print("respNewSettings=%s" % respNewSettings)
    return respNewSettings

```

所以settings等参数，也要参考对应文档：

- GitHub中的
 - [appium/settings.md at master · appium/appium](#)
- readthedocs中的
 - [Settings - appium](#)
 - [The Settings API - Appium](#)

其中的ScreenshotQuality的值，是来自苹果的XCTest中的定义：

- [XCTImageQuality - XCTest | Apple Developer Documentation](#)

以及另外相关的Capabilities

- [appium/caps.md at master · appium/appium](#)
- [Update Settings - Appium](#)

而wda中其他还有地方也是用到Appium的，比如：

Attribute - Appium

```

HTTP API Specifications
Endpoint
GET /session/:session_id/elements/:element_id/attribute/:name

```

对应着能看到python的wda中就是：

文件： /Users/limao/.pyenv/versions/3.8.0/Python.framework/Versions/3.8/lib/python3.8/site-packages/wda/__init__.py

```

def _wdasearch(self, using, value):
    """

```

```

    Returns:
        element_ids (list(string)): example ['id1', 'id2']

    HTTP example response:
    [
        {"ELEMENT": "E2FF5B2A-DBDF-4E67-9179-91609480D80A"},
        {"ELEMENT": "597B1A1E-70B9-4CBE-ACAD-40943B0A6034"}
    ]
    .....
    element_ids = []
    for v in self.http.post('/elements', {
        'using': using,
        'value': value
    }).value:
        element_ids.append(v['ELEMENT'])
    return element_ids

```

和：

文件： refer/WebDriverAgent/WebDriverAgentLib/Commands/FBFFindElementCommands.m

```

+ (NSArray *)routes
{
    return @{
        [[FBRoute POST @"/element"] respondWithTarget self action @selector(handleFindElement)],
        [[FBRoute POST @"/elements"] respondWithTarget self action @selector(handleFindElements)],
        [[FBRoute POST @"/element/:uuid/element"] respondWithTarget self action @selector(handleFindSubElement)],
        [[FBRoute POST @"/element/:uuid/elements"] respondWithTarget self action @selector(handleFindSubElements)],
        ...
    };
}

```

调试时开启debug后可以看到：

```

HAIN: **/XCUIElementTypeAny[`name == '通讯录'`]
Shell: curl -X POST -d '{"using": "class chain", "value": "**/XCUIElementTypeAny[`name == '\u0901a\u08baf\u05f55`"]"}' 'http://192.168.31.43:8100/session/C69F63F6-80EA-47DD-BA04-0A912945CE39/elements'
Return (984ms):
    "value" : [
        {
            "ELEMENT" : "15000000-0000-0000-0502-000000000000",
            "element-6066-11e4-a52e-4f735466cecf" : "15000000-0000-0000-0502-000000000000"
        }
    ],
    "sessionId" : "C69F63F6-80EA-47DD-BA04-0A912945CE39"
}
...

```

```
Shell: curl -X GET -d '' 'http://192.168.31.43:8100/session/C69F63F6-80EA-47DD-BA04-0A912945CE39/element/15000000-0000-0000-0502-000000000000/rect'
```

```
Return (688ms): {  
    "value" : {  
        "y" : 619,  
        "x" : 96,  
        "width" : 90,  
        "height" : 48  
    },  
    "sessionId" : "C69F63F6-80EA-47DD-BA04-0A912945CE39"  
}
```

去获取element的属性值的。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09 10:13:38

附录

下面列出相关参考资料。

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09 10:13:38

iOS内置app的bundle ID

用wda开发期间，常会涉及到操作app，其中需要知道内置app的bundle id。

下面是iOS 13自带的app的bundle ID：

App Name	Bundle ID
Activity	com.apple.Fitness
App Store	com.apple.AppStore
Apple Store	com.apple.store.Jolly
Books	com.apple.iBooks
Calculator	com.apple.calculator
Calendar	com.apple.mobilecal
Camera	com.apple.camera
Clips	com.apple.clips
Clock	com.apple.mobletimer
Compass	com.apple.compass
Contacts	com.apple.MobileAddressBook
FaceTime	com.apple.facetime
Files	com.apple.DocumentsApp
Find My	com.apple.findmy
GarageBand	com.apple.mobilegarageband
Health	com.apple.Health
Home	com.apple.Home
iCloud Drive	com.apple.iCloudDriveApp
iMovie	com.apple.iMovie
iTunes Store	com.apple.MobileStore
iTunes U	com.apple.itunesu
Mail	com.apple.mobilemail
Maps	com.apple.Maps
Messages	com.apple.MobileSMS
Measure	com.apple.measure
Music	com.apple.Music
News	com.apple.news

Notes	com.apple.mobilenotes
Phone	com.apple.mobilephone
Photos	com.apple.mobileslideshow
Photo Booth	com.apple.Photo-Booth
Podcasts	com.apple.podcasts
Reminders	com.apple.reminders
Safari	com.apple.mobilesafari
Settings	com.apple.Preferences
Shortcuts	com.apple.shortcuts
Stocks	com.apple.stocks
Tips	com.apple.tips
TV	com.apple.tv
Videos	com.apple.videos
Voice Memos	com.apple.VoiceMemos
Wallet	com.apple.Passbook
Watch	com.apple.Bridge
Weather	com.apple.weather

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新: 2020-08-09 10:13:38

文档

facebook的WebDriverAgent

[facebookarchive/WebDriverAgent: A WebDriver server for iOS that runs inside the Simulator.](#)

已archive归档，不更新了。

-》常见问题：

[Common Issues · facebookarchive/WebDriverAgent Wiki](#)

其中提到了：

关于中国的iPhone，无法通过IP访问的问题：

[Can't connect via Wifi to iOS10.x devices in China · Issue #288 · facebookarchive/WebDriverAgent](#)

USB的支持：

[USB support · facebookarchive/WebDriverAgent Wiki](#)

-》关于Inspector：

[Using the Inspector · facebookarchive/WebDriverAgent Wiki](#)

关于内部query查询api接口：

[Queries · facebookarchive/WebDriverAgent Wiki](#)

关于Predicate Queries：

[Predicate Queries Construction Rules · facebookarchive/WebDriverAgent Wiki](#)

apple的相关资料

[Predicate Format String Syntax](#)

如何提高搜索查找的速度

[How To Achieve The Best Lookup Performance · facebookarchive/WebDriverAgent Wiki](#)

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2020-08-09 10:13:38

参考资料

- 【已解决】Mac中用facebook-wda自动化测试操作iOS设备
- 【已解决】Mac中用facebook-wda操作iOS真机iPhone6
- 【已解决】Mac中xcodebuild警告：xcode-select error tool xcodebuild requires Xcode
- 【未解决】自动抓包iOS的app：左滑引导页进入首页
- 【已解决】自动抓包工具适配iOS：当前检测出微信连续异常，你可以尝试一下方法修复 下一步
- 【已解决】iPhone中AppStore的bundle id即appId是啥
- 【已解决】facebook-wda中元素clear清除文本值无效
- 【已解决】facebook-wda获取鉴定的value值是错误的
- 【未解决】自动抓包iOS的app：无忧筹点击首页的筹款首页后无法返回
- 【未解决】facebook-wda点击个人所得税元素无效：没有进入AppStore详情页
- 【未解决】研究facebook-wda和WebDriverAgent中attribute/value始终是null无法获取有效值
- 【未解决】WebDriverAgent获取iPhone页面源码报错：Code 5 Error kAXErrorIPCTimeout
getting snapshot for element
- 【已解决】自动抓包iOS公众号：niuer-tmall中丢失部分主菜单
- 【已解决】Python中facebook-wda和WebDriverAgent中是否可以支持displayed以及是否能替换visible
- 【已解决】自动抓包iOS公众号：niuer-tmall定位主菜单失败
- 【已解决】合并最新版WebDriverAgent后测试是否支持元素的visible属性的query查询
- 【未解决】自动抓包iOS公众号：小程序中可关闭弹框签到赢好礼
- 【规避解决】自动抓包iOS公众号：获取微信公众号unesunes搜索结果页面源码失败
- 【记录】更新WebDriverAgent后测试iOS 12的iPhone6抓包微信公众号
- 【规避解决】XCode实时调试NSXPConnection的
`_XCT_fetchSnapshotForElement:attributes:parameters:reply`错误
- 【规避解决】WebDriverAgent获取页面源码报错：`xpc.exceptions NSXPConnection com.apple.testmanagerd _XCT_fetchSnapshotForElement`
- 【已解决】wda用source()获取页面源码xml速度极其慢
- 【已解决】把旧版WebDriverAgent自己优化改动合并到最新版代码中
- 【已解决】验证最新WebDriverAgent代码功能上是否正常
- 【已解决】XCode编译最新版WebDriverAgent
- 【已解决】用XCode实时调试WebDriverAgent希望找到并解决获取页面源码慢的原因
- 【已解决】尝试解决facebook-wda和WebDriverAgent的获取源码很慢的原因
- 【未解决】WebDriverAgent和wda获取源码提速：尝试`shouldLoadSnapshotWithAttributes`参数
- 【未解决】调节Appium的Capability的参数去提高facebook-wda和WebDriverAgent获取源码的速度
- 【未解决】WebDriverAgent获取源码慢尝试调节参数：
`shouldUseTestManagerForVisibilityDetection`
- 【已解决】Xcode调试WebDriverAgent研究`fb_waitUntilSnapshotIsStable`含义希望提高获取源码速度
- 【已解决】WebDriverAgent报错：Internal error Error Domain com.apple.dt.xctest.automation-support.error Code 5 Error kAXErrorServerNotFound getting snapshot for element

- 【已解决】WebDriverAgent中fb_waitUntilSnapshotIsStable的作用和含义即为何加上
- 【已解决】WebDriverAgent获取源码慢尝试调节参数：FB_ANIMATION_TIMEOUT
- 【已解决】Mac中安装和初始化facebook-wda环境
- 【已解决】Mac中XCode如何编译WebDriverAgent.xcodeproj
- 【已解决】用XCode给WebDriverAgent.xcodeproj的WebDriverAgentRunner添加设置code signing代码签名
- 【已解决】Mac中XCode中WebDriverAgent编译报错： Signing for IntegrationApp requires a development team
- 【已解决】XCode中WebDriverAgent.xcodeproj自动签名报错： Failed to register bundle identifier com.facebook.WebDriverAgentRunner
- 【已解决】XCode 11中如何修改WebDriverAgentRunner的bundle identifier
-
- crifan (Crifan Li)
- 【记录】给Gitbook添加更多配置和功能
- 【已解决】提取Gitbook中Makefile公共部分
- 【已解决】gitbook中book.json中能否把公共部分提取出来
- 端口转发 · 苹果相关开发总结
- ATX 文档 - iOS 真机如何安装 WebDriverAgent · TesterHome
- ATX 系列-如何测试网易云音乐 (iOS 篇) · TesterHome
- 使用 Python 库 facebook-wda 完成网易云音乐 iOS 客户端的自动化测试 (示例) · TesterHome
- ATX 文档 - iOS 控件操作 API · TesterHome
- iOS 自动化测试 · TesterHome
- 【IOS测试】一篇读懂自动化框架WebDriverAgent – Python量化投资
- iOS真机安装WebDriverAgent | Vicの博客
-

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新：2021-04-11 10:23:14