

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
开发心得	1.4
iOS的各种坑	1.5
常用代码段	1.6
微信相关	1.6.1
弹框处理	1.6.2
首次登录引导页	1.6.3
源码分析	1.7
附录	1.8
文档	1.8.1
参考资料	1.8.2

iOS自动化测试利器：facebook-wda

- 最新版本: v1.1
- 更新时间: 20200620

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。

简介

总结iOS自动化主流框架facebook-wda的简介，环境搭建以及常见问题，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。

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：2020-06-20 23:12:18

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](#)

- 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 XCUI Test
 - 是什么：苹果的测试框架
 - 官网文档
 - 之前：
 - XCUI Test
 - (貌似) 最新
 - [XCTest | Apple Developer Documentation](#)
- 关于：Appium
 - [Appium: Mobile App Automation](#)

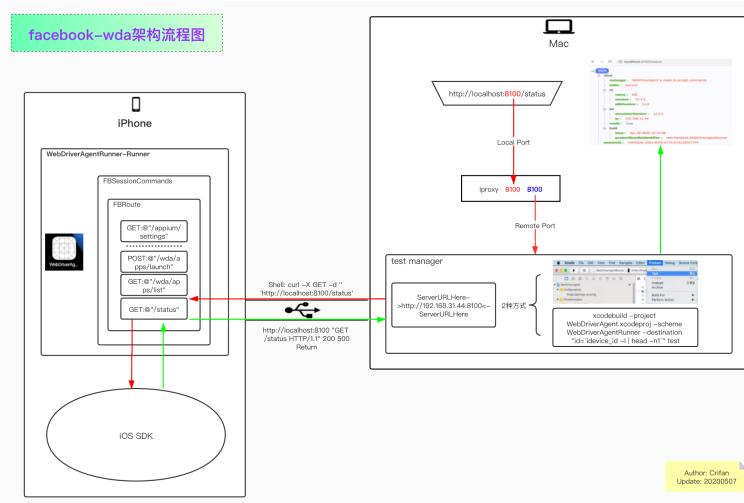
crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved,
powered by Gitbook最后更新：2020-06-01 18:33:14

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，需要先去给 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：



此处长按变待删除，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：

```
□ ll
total 96
-rw-r--r-- 1 limao CORP\Domain Users 2.0K 2 13 17:40
-rw-r--r-- 1 limao CORP\Domain Users 111B 2 13 17:40
-rw-r--r-- 1 limao CORP\Domain Users 75B 2 13 17:40
drwxr-xr-x 3 limao CORP\Domain Users 96B 2 13 17:40
drwxr-xr-x 4 limao CORP\Domain Users 128B 2 13 17:40
-rw-r--r-- 1 limao CORP\Domain Users 177B 2 13 17:40
-rw-r--r-- 1 limao CORP\Domain Users 4.3K 2 13 17:40
-rw-r--r-- 1 limao CORP\Domain Users 1.5K 2 13 17:40
-rw-r--r-- 1 limao CORP\Domain Users 1.9K 2 13 17:40
drwxr-xr-x 7 limao CORP\Domain Users 224B 2 13 17:40
-rw-r--r-- 1 limao CORP\Domain Users 2.5K 2 13 17:40
drwxr-xr-x 5 limao CORP\Domain Users 160B 2 13 17:40
drwxr-xr-x 5 limao CORP\Domain Users 160B 2 13 17:40
drwxr-xr-x 16 limao CORP\Domain Users 512B 2 13 17:40
drwxr-xr-x 4 limao CORP\Domain Users 128B 2 13 17:40
drwxr-xr-x 6 limao CORP\Domain Users 192B 2 13 17:40
drwxr-xr-x 5 limao CORP\Domain Users 160B 2 13 17:40
-rw-r--r-- 1 limao CORP\Domain Users 666B 2 13 17:40
-rw-r--r-- 1 limao CORP\Domain Users 896B 2 13 17:40
drwxr-xr-x 9 limao CORP\Domain Users 288B 2 13 17:40
-rw-r--r-- 1 limao CORP\Domain Users 2.5K 2 13 17:40
drwxr-xr-x 5 limao CORP\Domain Users 160B 2 13 17:40
```

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 -l | head -n1`"
test
```
 - 或分2步
 - 先获取iOS设备的UDID:

```
CUR_UDID=$(idevice_id -l |
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 -l` 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
2020-02-20 10:51:00.119667+0800 WebDriverAgentRunner-Runner
2020-02-20 10:51:00.123946+0800 WebDriverAgentRunner-Runner
```

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

crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved,
powered by Gitbook最后更新: 2020-06-02 15:11:33

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
```

- 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
xcodebuild: error: missing value for key 'id' of option 'D
```

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] IDETestAgent[2440]: Device: DNPND9S1G5MR  
deviceSerialNumber: DNPND9S1G5MR  
identifier: ed94089f3e34d5538065a69  
deviceClass: iPhone  
deviceName: Crifan iPhone6  
deviceIdentifier: ed94089f3e34d5538065a69  
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:0x7f8456c000000000  
supportedDeviceFamilies: ()  
  1  
)  
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::  
activeProxiedDevice: (null)  
} (12.4.5 (16G161))
```

USB端口转发

为了测试更方便，最好安装和启动端口转发

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
```

如果已端口转发则可以把IP换localhost

如果用了端口转发，则可以把IP换成localhost：

```
http://localhost: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.WebDriverAgent"  
        }  
    },  
    "sessionId": "38289A64-E467-4458-A0F1-8A3B2A6AAECE"  
}
```



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: 2020-06-18 10:05:52

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：

```
gWdaClient = wda.Client('http://localhost:8100')  
gWdaClient = wda.Client('http://localhost:8101')  
gWdaClient = 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系统封闭和不开放库苦不堪言，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: 2020-06-02 09:53:29

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，现在来详细说说：

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">
```

去用：

```
{'value': '可申请(元) 200,000', 'name': '可申请(元) 200,000',
```

以及 去掉y的：

```
{'value': '可申请(元) 200,000', 'name': '可申请(元) 200,000',
```

都查不到元素。

不过，去掉value, name, label后：

```
{'enabled': 'true', 'x': '85', 'y': '226', 'width': '244',
```

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">
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true">
        <XCUIElementTypeScrollView type="XCUIElementTypeScrollView">
            <XCUIElementTypeImage type="XCUIElementTypeImage" identifier="button">
                </XCUIElementTypeImage>
            <XCUIElementTypeButton type="XCUIElementTypeButton" identifier="button">
                <XCUIElementTypeText type="XCUIElementTypeText" value="立即进入">
                </XCUIElementTypeText>
            </XCUIElementTypeButton>
        </XCUIElementTypeScrollView>
    </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="1">
    <XCUIElementTypeTable type="XCUIElementTypeTable" name="Table">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="1">
            <XCUIElementTypeImage type="XCUIElementTypeImage" name="Image" value="Image" />
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="Text" />
            <XCUIElementTypeButton type="XCUIElementTypeButton" value="Button" />
        </XCUIElementTypeOther>
    </XCUIElementTypeTable>
</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="系统正在开小差再刷新下">
    <XCUIElementTypeOther type="XCUIElementTypeOther" name="系统正在开小差再刷新下">
        <XCUIElementTypeOther type="XCUIElementTypeOther" name="系统正在开小差再刷新下">
            <XCUIElementTypeImage type="XCUIElementTypeImage" name="系统正在开小差再刷新下">
                <XCUIElementTypeStaticText type="XCUIElementTypeText" name="系统正在开小差再刷新下">
                    </XCUIElementTypeText>
                </XCUIElementTypeImage>
            </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther" name="系统正在开小差再刷新下">
            <XCUIElementTypeOther type="XCUIElementTypeOther" name="系统正在开小差再刷新下">
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </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">
    <XCUIElementTypeWindow type="XCUIElementTypeWindow" enabled="true">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true">
                    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true">
                        </XCUIElementTypeOther>
                    </XCUIElementTypeOther>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeWindow>
        <XCUIElementTypeWindow type="XCUIElementTypeWindow" enabled="true">
            <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true">
                </XCUIElementTypeWindow>
            <XCUIElementTypeWindow type="XCUIElementTypeWindow" enabled="true">
                <XCUIElementTypeWindow type="XCUIElementTypeWindow" enabled="true">
                    <XCUIElementTypeStatusBar type="XCUIElementTypeStatusBar" enabled="true">
                </XCUIElementTypeWindow>
            </XCUIElementTypeWindow>
            <XCUIElementTypeWindow type="XCUIElementTypeWindow" enabled="true">
                <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true">
                    </XCUIElementTypeWindow>
                </XCUIElementTypeOther>
            </XCUIElementTypeWindow>
        </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">
    <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" />
    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="【善友筹】(已审核)" />
    <XCUIElementTypeProgressIndicator type="XCUIElementTypeProgressIndicator" value="14%" />
    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="烧香拜佛求平安,无助的眼神,身上插着数根管子。面对病魔无能为力,恳求好心人帮帮我们,日后有能力我们一定会多做善事,去帮助他人!" />
    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="14%" />
    <XCUIElementTypeTextCell type="XCUIElementTypeTextCell" value="发起人:刘杰" />
    <XCUIElementTypeTextCell type="XCUIElementTypeTextCell" value="认筹金额:2757" />
    <XCUIElementTypeTextCell type="XCUIElementTypeTextCell" value="剩余:20天" />
</XCUIElementTypeCell>
```

其中

```
<XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" />
```

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！

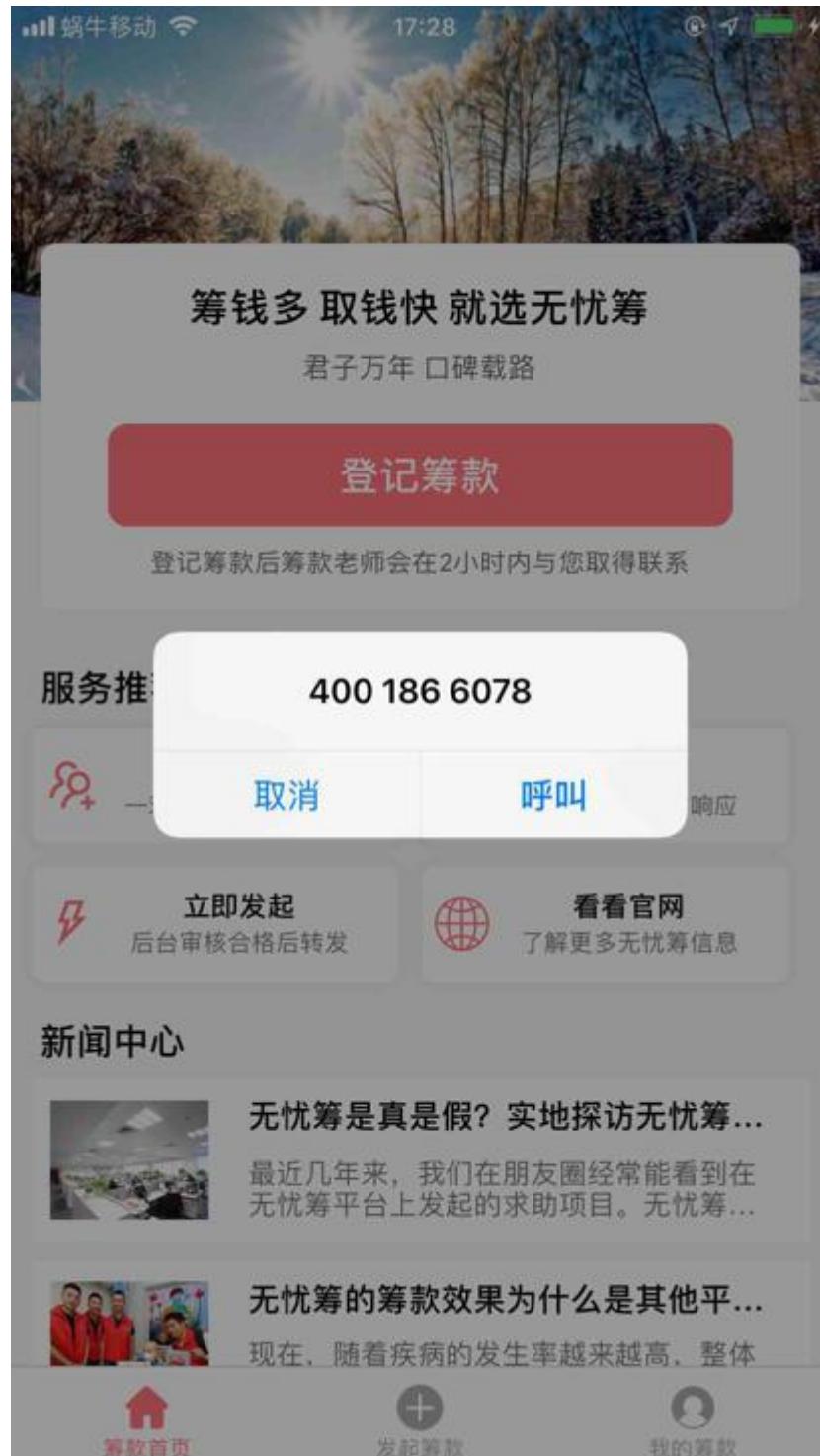
你妹的，那还怎么解析出有效节点，根本没法提取有效节点，和后续抓取。

详见：

【未解决】自动抓包工具抓包iOS的app：善友筹

坑：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：



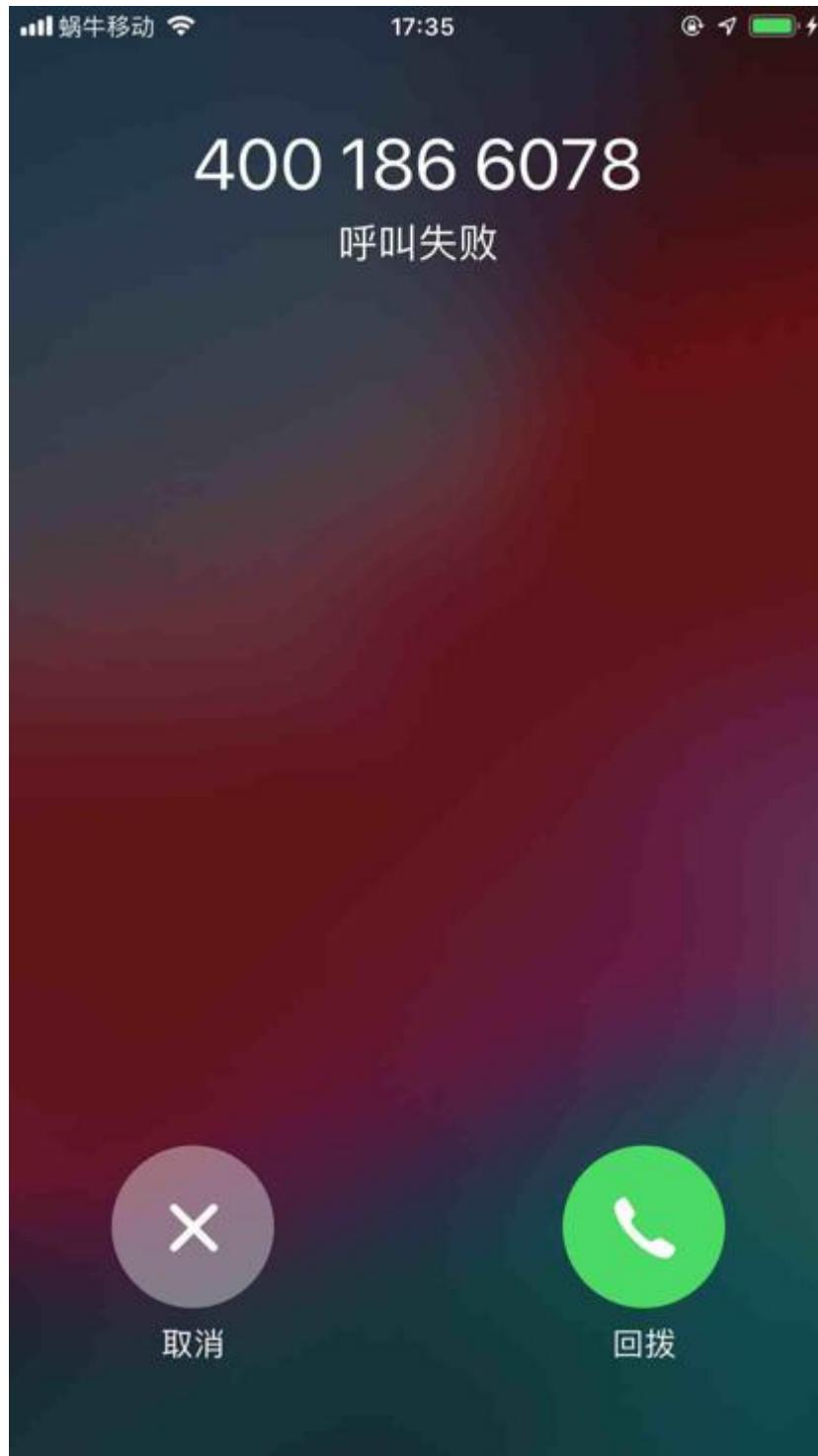
代码已找到了 取消 按钮，然后去点击 其中间坐标位置

```
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"})
#         )
#     if cancelSoup:
#         clickCenterPosition(curSession, cancelSoup)
#         foundAndProcessedPopup = True

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

cancelButtonQuery = {"type": "XCUIElementTypeButton"}
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。

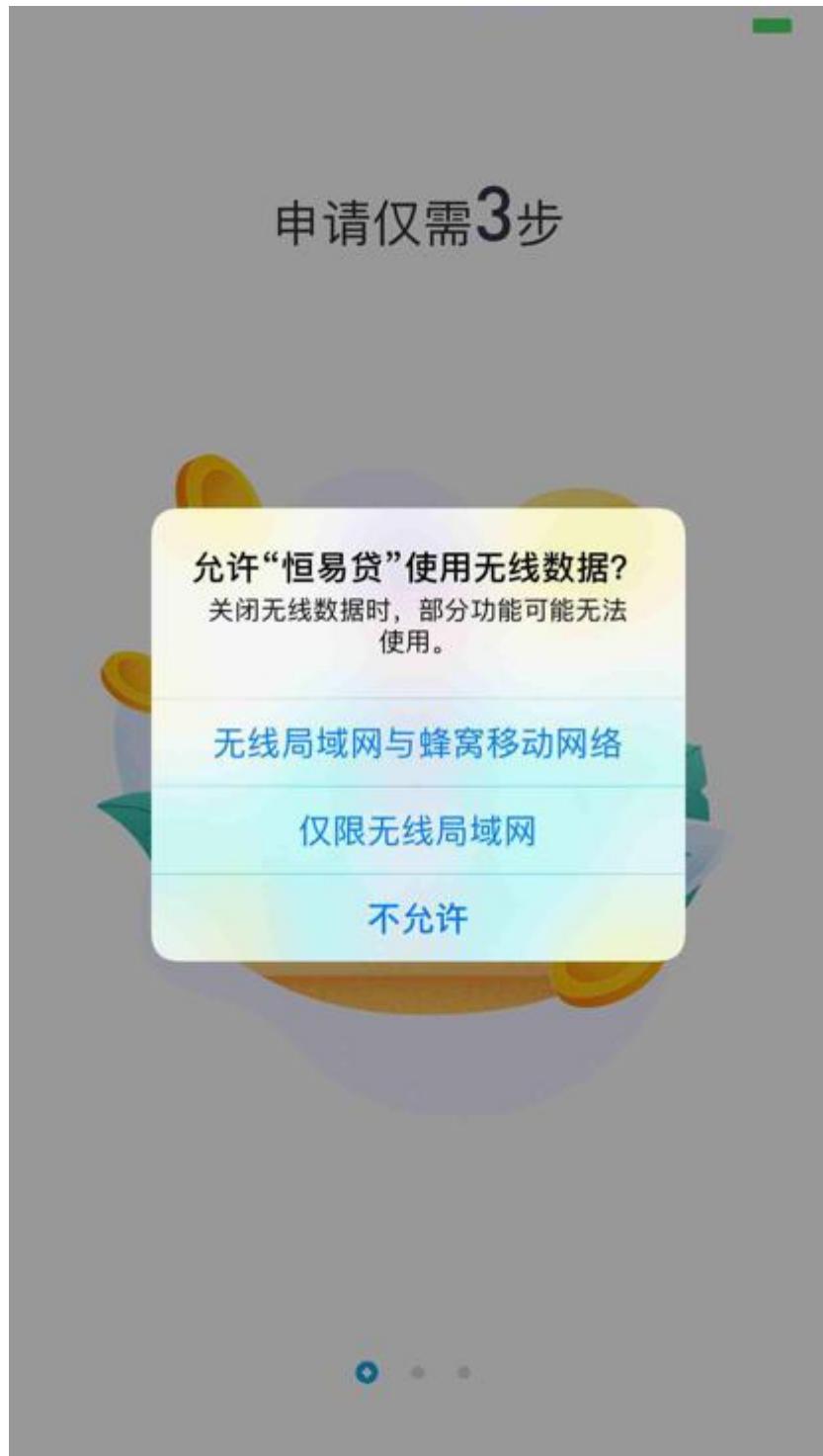
【后记2】

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": "蜂窝移动网络与无线局域网按钮中间位置"}
    },
]
wifiCellularSoup = utils.bsChainFind(soup, wifiCellularChainList)
if wifiCellularSoup:
    clickCenterPosition(curSession, wifiCellularSoup.attrs["name"])
    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
    # change to wda query element then click by element
    curName = wifiCellularSoup.attrs["name"] # 好
    wifiCellularButtonQuery = {"type": "XCUIElement", "name": curName}
    foundAndClicked = self.findAndClickElement(wifiCellularButtonQuery)
    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[常用代码段](#)中的 `findAndClickButtonElementBySoup`

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：

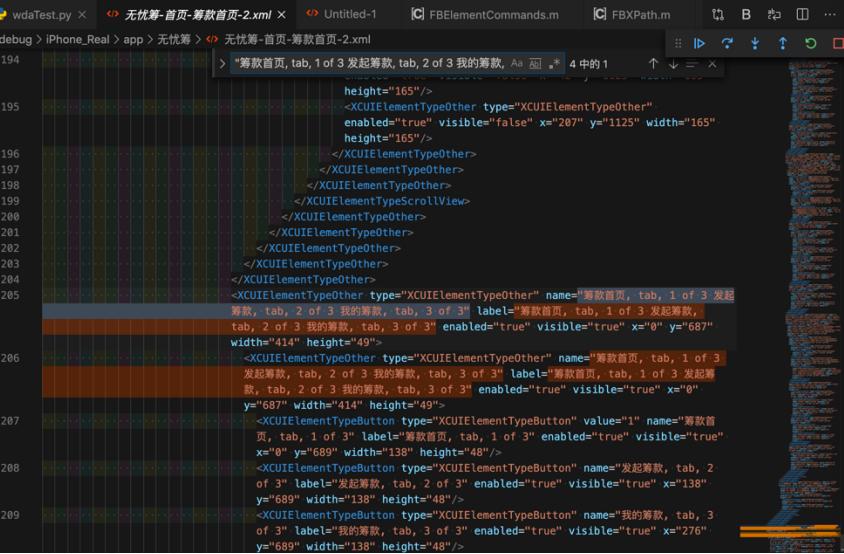


左下角的 3个tab页的父级元素，对应 locator，调试出现警告：

[200515 14:23:27] [ParsePage.py 1019] Found 2 same node from

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 我的筹款">
```



The screenshot shows a code editor with several tabs open. The active tab contains XML code for an iOS application's UI. The XML defines three tabs, each with a specific name and content. The names include "筹款首页" (Fundraising Home), "发起筹款" (Launch Fundraising), and "我的筹款" (My Fundraising). The XML structure is repetitive, with each tab having similar attributes like height, width, and position.

```
194 <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 我的筹款">
195   height="165"/>
196   -> XCUIElementTypeOther type="XCUIElementTypeOther"
197     enabled="true" visible="false" x="207" y="1125" width="165"
198     height="165"/>
199     -> XCUIElementTypeOther
200     -> XCUIElementTypeScrollView
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" 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" 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

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

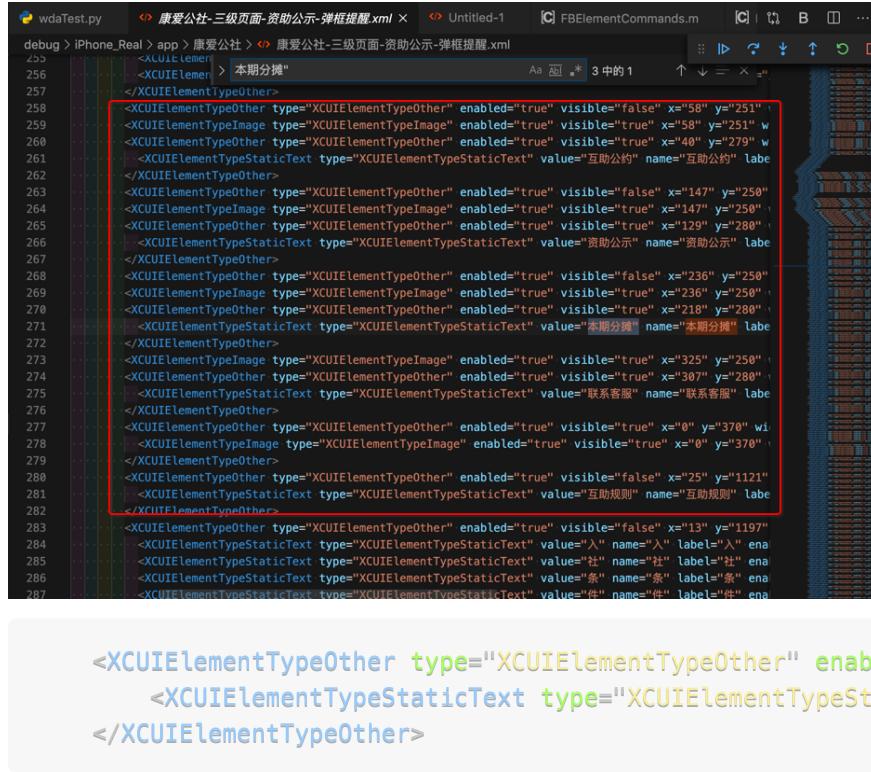


其中，正常的符合预期的是，页面xml源码中，有页面中的元素，比如顶部的第二排的 互助公约 资助公示 本期分摊 联系客服 等但是点击了 资助公示 后，进入 三级页面：
康爱公社-三级页面-资助公示-弹框提醒.jpg



竟然其中xml源码中，还有前一页的页面元素：

常见问题和心得



```
wdaTest.py  康爱公社-三级页面-资助公示-弹框提醒.xml  Untitled-1  FBElementCommands.m  C | t | B | ...  
debug > iPhone_Real > app > 康爱公社 > 康爱公社-三级页面-资助公示-弹框提醒.xml  
255     <XCUIElementTypeOther type="XCUIElementTypeImage" enabled="true" visible="false" x="58" y="251" w="100" h="100" label=""/>  
256     <XCUIElementTypeOther type="XCUIElementTypeImage" enabled="true" visible="true" x="58" y="251" w="100" h="100" label=""/>  
257     <XCUIElementTypeOther type="XCUIElementTypeImage" enabled="true" visible="true" x="40" y="279" w="100" h="100" label=""/>  
258     <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="互助公约" name="互助公约" label=""/>  
259     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="100" h="100" label=""/>  
260     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="100" h="100" label=""/>  
261     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="100" h="100" label=""/>  
262     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="100" h="100" label=""/>  
263     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="100" h="100" label=""/>  
264     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="100" h="100" label=""/>  
265     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="100" h="100" label=""/>  
266     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="100" h="100" label=""/>  
267     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="100" h="100" label=""/>  
268     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="100" h="100" label=""/>  
269     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="100" h="100" label=""/>  
270     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="100" h="100" label=""/>  
271     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="100" h="100" label=""/>  
272     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="100" h="100" label=""/>  
273     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="100" h="100" label=""/>  
274     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="100" h="100" label=""/>  
275     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="100" h="100" label=""/>  
276     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="100" h="100" label=""/>  
277     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="100" h="100" label=""/>  
278     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="100" h="100" label=""/>  
279     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="100" h="100" label=""/>  
280     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="100" h="100" label=""/>  
281     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="100" h="100" label=""/>  
282     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="100" h="100" label=""/>  
283     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="100" h="100" label=""/>  
284     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="100" h="100" label=""/>  
285     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="100" h="100" label=""/>  
286     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="100" h="100" label=""/>  
287     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="100" h="100" label=""/>  
  
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="13" y="1197" w="100" h="100" label=""/>  
    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="入" name="入" label=""/>  
    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="社" name="社" label=""/>  
    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="条" name="条" label=""/>  
    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="件" name="件" label=""/>
```

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。

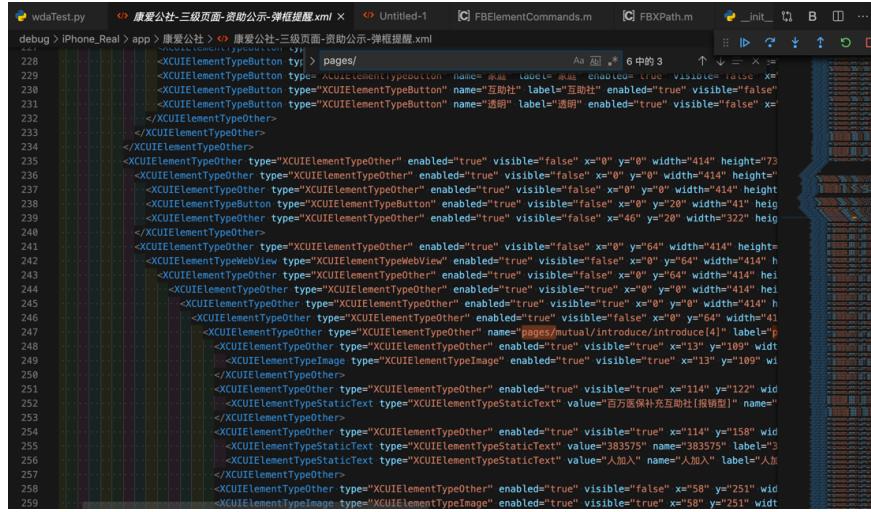
仔细去看xml源码中发现，有个特点：

会存在 pages/xxx/xxx 之类的元素：



```
<XCUIElementTypeOther type="XCUIElementTypeOther" name="pages" ...>  
  . . .
```

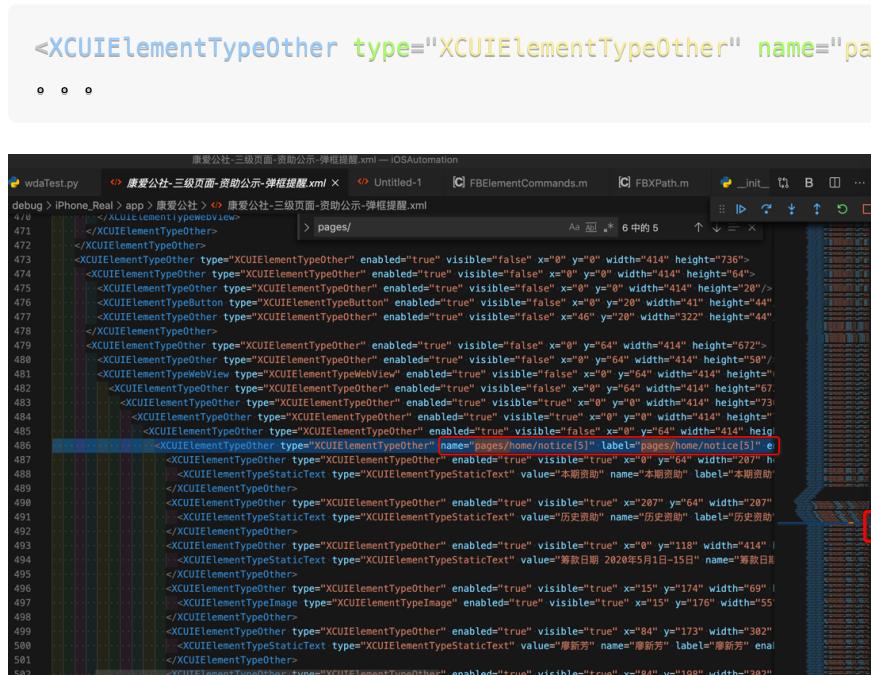
常见问题和心得



```
debug > iPhone_Real > app > 康爱公社 > 康爱公社-三级页面-资助公示-弹框提醒.xml
```

```
4/6 <XCUIElementTypeOther type="XCUIElementTypeOther" name="pages"><XCUIElementTypeButton type="XCUIElementTypeButton" name="康爱公社" label="康爱公社" enabled="true" visible="false" x="0" y="0" width="414" height="73"><XCUIElementTypeButton type="XCUIElementTypeButton" name="互助社" label="互助社" enabled="true" visible="false" x="0" y="0" width="414" height="73"><XCUIElementTypeButton type="XCUIElementTypeButton" name="透明" label="透明" enabled="true" visible="false" x="0" y="0" width="414" height="73"></XCUIElementTypeButton></XCUIElementTypeButton></XCUIElementTypeOther><XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="0" width="414" height="73"><XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="0" width="414" height="73"><XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="0" width="414" height="73"><XCUIElementTypeOther type="XCUIElementTypeButton" enabled="true" visible="false" x="0" y="28" width="414" height="414"><XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="46" y="28" width="322" height="414"></XCUIElementTypeOther></XCUIElementTypeOther></XCUIElementTypeOther><XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="44" width="414" height="73"><XCUIElementTypeOther type="XCUIElementTypeWebview" type="XCUIElementTypeWebview" enabled="true" visible="false" x="0" y="64" width="414" height="73"><XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="64" width="414" height="73"><XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="414"><XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="64" width="414" height="414"><XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="414"><XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="64" width="414" height="414"><XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="414"><XCUIElementTypeOther type="XCUIElementTypeOther" name="pages/mutual/introduce/introduce[4]" label="介绍[4]" enabled="true" visible="true" x="13" y="109" width="414" height="414"><XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="13" y="109" width="414" height="414"><XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="114" y="122" width="414" height="414"><XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="百万医脉补充互助社(报销型)" name="百万医脉补充互助社(报销型)" label="百万医脉补充互助社(报销型)" enabled="true" visible="true" x="114" y="122" width="414" height="414"><XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="58" y="251" width="414" height="414"><XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="加入" name="加入" label="加入" enabled="true" visible="true" x="58" y="251" width="414" height="414"><XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="58" y="251" width="414" height="414">
```

且不止一个：



```
debug > iPhone_Real > app > 康爱公社 > 康爱公社-三级页面-资助公示-弹框提醒.xml
```

```
4/6 <XCUIElementTypeOther type="XCUIElementTypeOther" name="pages"><XCUIElementTypeWebview type="XCUIElementTypeWebview" 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="64"><XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="0" width="414" height="20"/><XCUIElementTypeOther type="XCUIElementTypeButton" enabled="true" visible="false" x="0" y="20" width="414" height="44"><XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="46" y="20" width="322" height="44"></XCUIElementTypeOther><XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="64" width="414" height="672"><XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="64" width="414" height="58"/><XCUIElementTypeWebview type="XCUIElementTypeWebview" enabled="true" visible="false" x="0" y="64" width="414" height="58"><XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="64" width="414" height="67"/><XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="64" width="414" height="73"/><XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="64" width="414" height="16"/><XCUIElementTypeOther type="XCUIElementTypeOther" name="pages/home/notice[5]" label="pages/home/notice[5]" enabled="true" visible="true" x="0" y="64" width="414" height="16"><XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="64" width="207" height="16"><XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="本期资助" name="本期资助" label="本期资助" enabled="true" visible="true" x="0" y="64" width="207" height="16"/><XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="207" y="64" width="207" height="16"/><XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="历史资助" name="历史资助" label="历史资助" enabled="true" visible="true" x="207" y="64" width="207" height="16"/><XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="207" y="118" width="207" height="16"/><XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="筹款日期 2020年5月1-15日" name="筹款日期" label="筹款日期" enabled="true" visible="true" x="207" y="118" width="207" height="16"/><XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="15" y="174" width="69" height="16"/><XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="15" y="174" width="69" height="16"/><XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="84" y="173" width="30" height="16"/><XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="摩新芳" name="摩新芳" label="摩新芳" enabled="true" visible="true" x="84" y="173" width="30" height="16"/><XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="84" y="198" width="30" height="16"/>
```

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

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：2020-06-02 10:32:46

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，供需要的参考：

findAndClickButtonElementBySoup

```

def findAndClickButtonElementBySoup(self, curButtonSoup):
    """
        iOS的bug: 根据bs找到了soup元素(往往是一个button)后
        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
            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",
    # # foundAndClicked = self.findAndClickElement(curButtonQuery)

    curButtonQuery = {"type": "XCUIElementTypeButton", "name": "OK"}
    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',
        else:
            # merge query
            # curButtonQuery = {**curButtonQuery, **extraQuery}
            curButtonQuery.update(extraQuery) # {'enabled': 'true', 'height': '32'}

    foundAndClicked = self.findAndClickElement(curButtonQuery)
    return foundAndClicked

```

不同的地方的各种调用：

```
foundAndProcessedPopup = self.findAndClickButtonElementByS
```

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):
    """use BeautifulSoup chain list to find soup node then click it

    Args:
        bsChainList (list): dict list for dict of tag and attr
        soup (Soup): BeautifulSoup soup
        isUseWdaQueryAndClick (bool): for special node bs click
    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.findAndClickButtonElementByS
        else:
            self.clickElementCenterPosition(foundSoup)
            foundAndClicked = True
    return foundAndClicked
```

场景：米家 弹框 立即体验 xml

```
<XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true">
    <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true">
        <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true">
            <XCUIElementTypeImage type="XCUIElementTypeImage" name="立即体验" value="立即体验" x="480" y="180" width="100" height="100"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeText" value="立即体验" x="480" y="180" width="100" height="100"/>
            <XCUIElementTypeStaticText type="XCUIElementTypeText" value="立即体验" x="480" y="180" width="100" height="100"/>
            <XCUIElementTypeButton type="XCUIElementTypeButton" value="立即体验" x="480" y="180" width="100" height="100"/>
        </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", "value": "立即体验", "x": 480, "y": 180, "width": 100, "height": 100}
    }
]
foundAndProcessedPopup = self.findAndClickCenterPosition(immediatelyExperienceChainList)
return foundAndProcessedPopup
```

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

```

def findAndClickButtonElementBySoup(self, curButtonSoup=None):
    """
        iOS的bug：根据bs找到了soup元素（往往是一个button）后，用
        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", "enab...
    # # foundAndClicked = self.findAndClickElement(curButtonQuery)

    curButtonQuery = {"type": "XCUIElementTypeButton", "enab...
    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': 'XCUIEl...
        else:
            # merge query
            # curButtonQuery = {**curButtonQuery, **extraQuery}
            curButtonQuery.update(extraQuery) # {'enabled': 'true', ...

    foundAndClicked = self.findAndClickElement(curButtonQuery)
    return foundAndClicked

```

说明：

- iOS的bug：根据bs找到了soup元素（往往是一个button）后，用clickCenterPosition=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, cor
if commonCloseSoup:
    # self.clickElementCenterPosition(commonCloseSo
    # foundAndProcessedPopup = True

    # so change to wda query element then click
foundAndProcessedPopup = self.findAndClickButton
```

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="XCUIElementType">
            <XCUIElementTypeOther type="XCUIElementType">
                <XCUIElementTypeButton type="XCUIElementType">
                    ...
                </XCUIElementTypeButton>
            </XCUIElementTypeOther>
        <XCUIElementTypeNavigationBar type="XCUIElementType">
            <XCUIElementTypeOther type="XCUIElementType">
                <XCUIElementTypeButton type="XCUIElementType">
                    ...
                </XCUIElementTypeButton>
            </XCUIElementTypeOther>
        <XCUIElementTypeNavigationBar type="XCUIElementType">
            <XCUIElementTypeButton type="XCUIElementType">
                <XCUIElementTypeOther type="XCUIElementType">
                    <XCUIElementTypeButton type="XCUIElementType">
                        ...
                    </XCUIElementTypeButton>
                </XCUIElementTypeOther>
            </XCUIElementTypeButton>
        </XCUIElementTypeNavigationBar>
   某个公众号: 动卡空间
        <XCUIElementTypeNavigationBar type="XCUIElementType">
            <XCUIElementTypeButton type="XCUIElementType">
                <XCUIElementTypeOther type="XCUIElementType">
                    <XCUIElementTypeButton type="XCUIElementType">
                        ...
                    </XCUIElementTypeButton>
                </XCUIElementTypeOther>
            </XCUIElementTypeButton>
        </XCUIElementTypeNavigationBar>
    not:
        <XCUIElementTypeNavigationBar type="XCUIElementType">
            <XCUIElementTypeButton type="XCUIElementType">
                <XCUIElementTypeOther type="XCUIElementType">
                    <XCUIElementTypeButton type="XCUIElementType">
                        ...
                    </XCUIElementTypeButton>
                </XCUIElementTypeOther>
            </XCUIElementTypeButton>
        </XCUIElementTypeNavigationBar>
    .....
    soup = CommonUtils.xmlToSoup(page)
    foundNavBar = soup.find(
        'XCUIElementTypeNavigationBar',
        attrs={"type": "XCUIElementTypeNavigationBar", "end": "1"})
    if foundNavBar:
        # maybeFakeNavBarName = foundNavBar.attrs["name"]
        maybeFakeNavBarName = foundNavBar.attrs.get("name")
        typeOtherNameP = re.compile("%s,?" % maybeFakeNavBarName)
        foundTypeOther = foundNavBar.find("XCUIElementTypeOther")
        if foundTypeOther:
            hasNavBar = True
            naviBarName = maybeFakeNavBarName
```

```
return hasNavBar, naviBarName
```

调用：

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

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：2020-06-20 22:49:28

微信相关

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
    isPublicAccountSearch = False
    curInputValue = ""

    .....
    is search:
        <XCUIElementTypeImage type="XCUIElementTypeImage">
            <XCUIElementTypeButton type="XCUIElementTypeButton">
                <XCUIElementTypeOther type="XCUIElementTypeOther">
                    <XCUIElementTypeOther type="XCUIElementTypeImage">
                        <XCUIElementTypeImage type="XCUIElementTypeImage">
                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                <XCUIElementTypeSearchField type="XCUIElementTypeSearchField">
                                    <XCUIElementTypeButton type="XCUIElementTypeButton">
                                        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
                                            </XCUIElementTypeButton>
                                        </XCUIElementTypeStaticText>
                                    </XCUIElementTypeButton>
                                </XCUIElementTypeOther>
                            </XCUIElementTypeSearchField>
                        </XCUIElementTypeImage>
                    </XCUIElementTypeOther>
                </XCUIElementTypeImage>
            </XCUIElementTypeButton>
        </XCUIElementTypeImage>
    </XCUIElementTypeImage>
    <XCUIElementTypeImage type="XCUIElementTypeImage">
        <XCUIElementTypeButton type="XCUIElementTypeButton">
            <XCUIElementTypeOther type="XCUIElementTypeOther">
                <XCUIElementTypeOther type="XCUIElementTypeImage">
                    <XCUIElementTypeImage type="XCUIElementTypeImage">
                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                            <XCUIElementTypeSearchField type="XCUIElementTypeSearchField">
                                <XCUIElementTypeButton type="XCUIElementTypeButton">
                                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
                                        </XCUIElementTypeButton>
                                    </XCUIElementTypeStaticText>
                                </XCUIElementTypeButton>
                            </XCUIElementTypeSearchField>
                        </XCUIElementTypeImage>
                    </XCUIElementTypeOther>
                </XCUIElementTypeImage>
            </XCUIElementTypeButton>
        </XCUIElementTypeImage>
    </XCUIElementTypeImage>
    <XCUIElementTypeImage type="XCUIElementTypeImage">
        <XCUIElementTypeButton type="XCUIElementTypeButton">
            <XCUIElementTypeOther type="XCUIElementTypeOther">
                <XCUIElementTypeOther type="XCUIElementTypeImage">
                    <XCUIElementTypeImage type="XCUIElementTypeImage">
                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                            <XCUIElementTypeSearchField type="XCUIElementTypeSearchField">
                                <XCUIElementTypeButton type="XCUIElementTypeButton">
                                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
                                        </XCUIElementTypeButton>
                                    </XCUIElementTypeStaticText>
                                </XCUIElementTypeButton>
                            </XCUIElementTypeSearchField>
                        </XCUIElementTypeImage>
                    </XCUIElementTypeOther>
                </XCUIElementTypeImage>
            </XCUIElementTypeButton>
        </XCUIElementTypeImage>
    </XCUIElementTypeImage>
```

```
        </XCUIElementTypeButton>
        </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    </XCUIElementTypeOther>
    </XCUIElementTypeImage>
    .....
soup = CommonUtils.xmlToSoup(page)
widthStr = str(self.X)
foundImage = soup.find(
    'XCUIElementTypeImage',
    attrs={"type": "XCUIElementTypeImage", "enabled": "true"})
if foundImage:
    foundSearchField = foundImage.find("XCUIElementTypeSearchField",
        attrs={"type": "XCUIElementTypeSearchField", "isPlaceholder": "false"})
    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 into
    isPublicAccountFocusOrInto = False

    .....

    is Focus:
        <XCUIElementTypeNavigationBar type="XCUIElementType">
            <XCUIElementTypeButton type="XCUIElementType">
                <XCUIElementTypeOther type="XCUIElementType">
                    <XCUIElementTypeButton type="XCUIElementType">
                        <XCUIElementTypeButton type="XCUIElementType">
                            <XCUIElementTypeButton type="XCUIElementType">
                                <XCUIElementTypeNavigationBar>

                                <XCUIElementTypeTable type="XCUIElementType">
                                    <XCUIElementTypeButton type="XCUIElementType">
                                        <XCUIElementTypeStaticText type="XCUIElementType">
                                            <XCUIElementTypeImage type="XCUIElementType">
                                                <XCUIElementTypeButton type="XCUIElementType">
                                                    <XCUIElementTypeStaticText type="XCUIElementType">
                                                        <XCUIElementTypeButton type="XCUIElementType">
                                                            <XCUIElementTypeStaticText type="XCUIElementType">
                                                                </XCUIElementTypeButton>
                                                                <XCUIElementTypeStaticText type="XCUIElementType">
                                                                    <XCUIElementTypeButton type="XCUIElementType">
                                                                        <XCUIElementTypeStaticText type="XCUIElementType">
                                                                            <XCUIElementTypeButton type="XCUIElementType">
                                                                                <XCUIElementTypeStaticText type="XCUIElementType">
                                                                                    </XCUIElementTypeButton>
                                                                                    <XCUIElementTypeStaticText type="XCUIElementType">
                                                                                        <XCUIElementTypeButton type="XCUIElementType">
                                                                                            <XCUIElementTypeStaticText type="XCUIElementType">
                                                                                                </XCUIElementTypeButton>
                                                                                                <XCUIElementTypeButton type="XCUIElementType">
                                                                                                    <XCUIElementTypeStaticText type="XCUIElementType">
                                                                                                        </XCUIElementTypeButton>

```

is enter into:

```
<XCUIElementTypeNavigationBar type="XCUIElementType">
    <XCUIElementTypeButton type="XCUIElementType">
        <XCUIElementTypeOther type="XCUIElementType">
            <XCUIElementTypeButton type="XCUIElementType">
                <XCUIElementTypeButton type="XCUIElementType">
                    <XCUIElementTypeNavigationBar>

                    <XCUIElementTypeTable type="XCUIElementType">
                        <XCUIElementTypeButton type="XCUIElementType">
                            <XCUIElementTypeStaticText type="XCUIElementType">
                                <XCUIElementTypeImage type="XCUIElementType">
                                    <XCUIElementTypeButton type="XCUIElementType">
                                        <XCUIElementTypeStaticText type="XCUIElementType">
                                            <XCUIElementTypeButton type="XCUIElementType">
                                                <XCUIElementTypeStaticText type="XCUIElementType">
                                                    </XCUIElementTypeButton>
                                                    <XCUIElementTypeStaticText type="XCUIElementType">
                                                        <XCUIElementTypeButton type="XCUIElementType">
                                                            <XCUIElementTypeStaticText type="XCUIElementType">
                                                                </XCUIElementTypeButton>
```

```

<XCUIElementTypeButton type="XCUIElement"
    <XCUIElementTypeStaticText type="XCUIElement"
        </XCUIElementTypeButton>
    <XCUIElementTypeButton type="XCUIElement"
        <XCUIElementTypeStaticText type="XCUIElement"
            </XCUIElementTypeButton>
        </XCUIElementTypeButton>

.....
soup = CommonUtils.xmlToSoup(page)
foundNavBar = soup.find(
    'XCUIElementTypeNavigationBar',
    attrs={"type": "XCUIElementTypeNavigationBar", "enabled": "true"})
# logging.debug("foundNavBar=%s", foundNavBar)
if foundNavBar:
    foundTypeOther = foundNavBar.find("XCUIElementType")
    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"})
            logging.debug("foundIntoAccount=%s", foundIntoAccount)
            if foundIntoAccount:
                prevSiblingList = foundIntoAccount.prevSiblings
                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['type']
                        curName = eachPrevSibling.attrs['name']
                        if (curType == TypeStaticText):
                            isPublicAccountFocusOrIntoAccount = True
                        else:
                            logging.debug("eachPrevSibling=%s", eachPrevSibling)

```

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

调用：

```
def isCurPageInOffline_iOS(self, page):
    """Check whether current page is belong to (unexcepted) 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.isPublicAccountSearch_iOS()
        if isPublicAccountSearch:
            # is in public account search page
            isOffline = True

    if not isOffline:
        isPublicAccountFocusOrIntoPage = self.isPublicAccountFocusOrIntoPage_iOS()
        if isPublicAccountFocusOrIntoPage:
            # is in public account focus or enter into page
            isOffline = True

    return isOffline
```

当发生异常提示时，一直点击（多数都是下一步），直到退出异常提示

```
def iOSMakesureIntoWeixin(self):
    """Makesure 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.iOSWeixinException()
            if foundAndProcessedException:
                beInWeixin = self.iOSIsInWeixin()

        maxRetryNum -= 1

    return beInWeixin

def iOSWeixinExceptionNextStep(self):
    foundAndClicked = False

    scrollViewClassChain = "/XCUIElementTypeScrollView[`re
        ButtonLabelNextStep = "下一步"
        ButtonLabelIntoWeixin = "进入微信"

        # nextStepButtonQuery = {"type":"XCUIElementTypeButton"}
        # nextStepButtonQuery["parent_class_chains"] = [scrollViewClassChain]

        # intoWeixinButtonQuery = {"type":"XCUIElementTypeButton"}
        # intoWeixinButtonQuery["parent_class_chains"] = [scrollViewClassChain]

        """
        <XCUIElementTypeScrollView type="XCUIElementTypeScrollView">
            <XCUIElementTypeImage type="XCUIElementTypeImage">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
                    <XCUIElementTypeButton type="XCUIElementTypeButton">
                        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
                            </XCUIElementTypeButton>
                        </XCUIElementTypeStaticText>
                    </XCUIElementTypeButton>
                </XCUIElementTypeStaticText>
            </XCUIElementTypeImage>
        </XCUIElementTypeScrollView>
        """

    # continuousExceptionQuery = {"type":"XCUIElementTypeStaticText"}
    # continuousExceptionQuery["parent_class_chains"] = [scrollViewClassChain]
    # isFoundContinuousException, respInfo = self.findElement(continuousExceptionQuery)
    # if isFoundContinuousException:
    #     isFoundNextStep, respInfo = self.findElement(nextStepButtonQuery)
    #     if isFoundNextStep:
    #         nextStepElement = respInfo
```

```
#         foundAndClicked = self.clickElement(nextStepElement)

.....
<XCUIElementTypeScrollView type="XCUIElementTypeScrollView">
<XCUIElementTypeImage type="XCUIElementTypeImage">
<XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
<XCUIElementTypeButton type="XCUIElementTypeButton">
    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
    </XCUIElementTypeButton>
<XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
<XCUIElementTypeButton type="XCUIElementTypeButton">
    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
    </XCUIElementTypeButton>
<XCUIElementTypeOther type="XCUIElementTypeOther">
<XCUIElementTypeOther type="XCUIElementTypeOther">
</XCUIElementTypeScrollView>
.....
# # if not foundAndClicked:
# rebootWeixinQuery = {"type":"XCUIElementTypeStaticText"}
# rebootWeixinQuery["parent_class_chains"] = [scrollView]
# isFoundRebootWeixin, respInfo = self.findElement(query)
# if isFoundRebootWeixin:
#     isFoundNextStep, respInfo = self.findElement(query)
#     if isFoundNextStep:
#         nextStepElement = respInfo
#         foundAndClicked = self.clickElement(nextStepElement)

.....
<XCUIElementTypeScrollView type="XCUIElementTypeScrollView">
<XCUIElementTypeOther type="XCUIElementTypeOther">
    <XCUIElementTypeImage type="XCUIElementTypeImage">
        <XCUIElementTypeImage type="XCUIElementTypeImage">
    </XCUIElementTypeOther>
<XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
<XCUIElementTypeButton type="XCUIElementTypeButton">
    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
    </XCUIElementTypeButton>
<XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
<XCUIElementTypeButton type="XCUIElementTypeButton">
    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
    </XCUIElementTypeButton>
<XCUIElementTypeOther type="XCUIElementTypeOther">
<XCUIElementTypeOther type="XCUIElementTypeOther">
</XCUIElementTypeScrollView>
.....
# # if not foundAndClicked:
# clearCacheQuery = {"type":"XCUIElementTypeStaticText"}
# clearCacheQuery["parent_class_chains"] = [scrollView]
# isFoundClearCache, respInfo = self.findElement(query)
# if isFoundClearCache:
#     isFoundNextStep, respInfo = self.findElement(query)
```

```
#     if isFoundNextStep:
#         nextStepElement = respInfo
#         foundAndClicked = self.clickElement(nextStepElement)

#####
<XCUIElementTypeScrollView type="XCUIElementTypeScrollView">
    <XCUIElementTypeImage type="XCUIElementTypeImage">
        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
            <XCUIElementTypeButton type="XCUIElementTypeButton">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
                    </XCUIElementTypeButton>
                <XCUIElementTypeButton type="XCUIElementTypeButton">
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
                        </XCUIElementTypeButton>
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
                        <XCUIElementTypeButton type="XCUIElementTypeButton">
                            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
                                </XCUIElementTypeButton>
                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                    </XCUIElementTypeScrollView>
#####
<XCUIElementTypeScrollView type="XCUIElementTypeScrollView">
    <XCUIElementTypeImage type="XCUIElementTypeImage">
        <XCUIElementTypeButton type="XCUIElementTypeButton">
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
                <XCUIElementTypeButton type="XCUIElementTypeButton">
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
                        </XCUIElementTypeButton>
                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                            </XCUIElementTypeScrollView>
#####
<XCUIElementTypeScrollView type="XCUIElementTypeScrollView">
    <XCUIElementTypeImage type="XCUIElementTypeImage">
        <XCUIElementTypeButton type="XCUIElementTypeButton">
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
                <XCUIElementTypeButton type="XCUIElementTypeButton">
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
                        </XCUIElementTypeButton>
                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                            </XCUIElementTypeScrollView>
#####
EachStepNoticeList = [
    # ("当前检测出微信连续异常，你可以尝试以下方法修复：", ButtonLabelINextStep),
    # ("当前检测出微信连续异常", ButtonLabelINextStep),
    # ("建议你重启手机，避免微信启动异常", ButtonLabelINextStep),
    # ("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", ButtonLabelINextStep),
    # ("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", ButtonLabelINextStep),
    # ("如果使用过程中还出现问题，建议你重启手机，更新系统，或者联系客服", ButtonLabelINextStep)
]
for (curStepNotice, buttonLabel) in EachStepNoticeList:
    # curNoticeQuery = {"type": "XCUIElementTypeStaticText", "label": curStepNotice}
    # curNoticeQuery = {"type": "XCUIElementTypeStaticText", "label": curStepNotice}
    curNoticeQuery = {"type": "XCUIElementTypeStaticText", "label": curStepNotice}
    curNoticeQuery["parent_class_chains"] = [scrollView]
    isFoundCurNotice, respInfo = self.findElement(query=curNoticeQuery)
    if isFoundCurNotice:
```

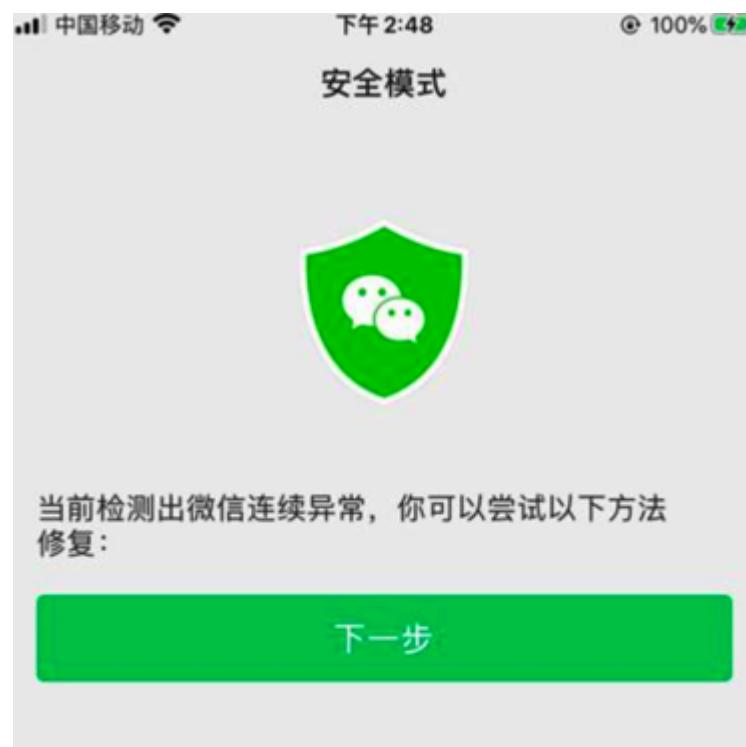
```
# isFoundNextStep, respInfo = self.findElement()
# isFoundButton, respInfo = self.findElement(query)
buttonQuery = {"type": "XCUIElementTypeButton",
               "buttonQuery": {"parent_class_chains": [scrollView]}}
isFoundButton, respInfo = self.findElement(query)
if isFoundButton:
    buttonElement = respInfo
    clickOk = self.clickElement(buttonElement)
    if clickOk:
        foundAndClicked = clickOk

return foundAndClicked
```

说明：

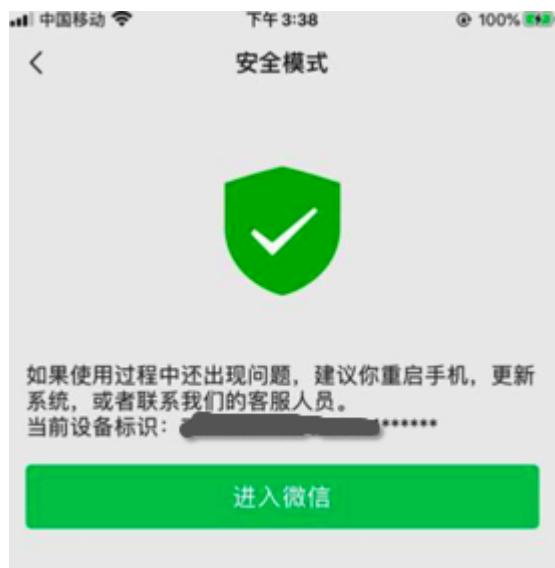
通过wda多次调试iOS微信，会导致退出再打开，时不时的会出现异常提示

希望一直点击，直到提示消失









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

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：2020-06-20 23:10:32

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.iOSProcessPopupUpperLeft()
    # if not foundAndProcessedPopup:
    #     foundAndProcessedPopup = self.iOSProcessPopupUpperRight()
    # Note: put common upper right close, for some app, eg
    if not foundAndProcessedPopup:
        # foundAndProcessedPopup = self.iOSProcessPopupUpperRight()
        foundAndProcessedPopup = self.iOSProcessPopupCommon()

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupPossibly()

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupUnder()

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupIKnow()

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessCommonUser()

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessUserPrivacy()

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessAlertUseWith()

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessCommonAlert()

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessCommonAlert()

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessRightUpper()

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupCommon()

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupCommon()
```

```
if not foundAndProcessedPopup:  
    foundAndProcessedPopup = self.iOSProcessPopupImmed:  
  
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_Fi

    if self.isSpecialiOSApp_kags:
        if not foundAndProcessedPopup:
            foundAndProcessedPopup = self.iOSProcessKagsPop

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupPhone

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessSettingsAll

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupCertifi

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupNetwo

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupNetwo

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupSyster

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupServer

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupWeixin

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupJumpT

    if not foundAndProcessedPopup:
        # foundAndProcessedPopup = self.iOSProcessPopupGet]
        foundAndProcessedPopup = self.iOSProcessPopupDoSome

    if not foundAndProcessedPopup:
        # foundAndProcessedPopup = self.iOSProcessPopupMin:
        foundAndProcessedPopup = self.iOSProcessPopupMinipi

    if not foundAndProcessedPopup:
        foundAndProcessedPopup = self.iOSProcessPopupWeixin
```

```
if not foundAndProcessedPopup:  
    foundAndProcessedPopup = self.iOSProcessPopupMinip  
  
if not foundAndProcessedPopup:  
    foundAndProcessedPopup = self.iOSProcessPopupMinip  
  
if not foundAndProcessedPopup:  
    foundAndProcessedPopup = self.iOSProcessWillOpenNot  
  
if not foundAndProcessedPopup:  
    foundAndProcessedPopup = self.iOSProcessAlertNoteC  
  
if not foundAndProcessedPopup:  
    foundAndProcessedPopup = self.iOSProcessAlertCance  
  
if not foundAndProcessedPopup:  
    foundAndProcessedPopup = self.iOSProcessAlertCommon  
  
if not foundAndProcessedPopup:  
    foundAndProcessedPopup = self.iOSProcessPopupPhoto  
  
# 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.iOSProcessCommonPop  
  
if not foundAndProcessedPopup:  
    foundAndProcessedPopup = self.iOSProcessPopupSover  
  
if not foundAndProcessedPopup:  
    foundAndProcessedPopup = self.iOSProcessPopupCommon  
  
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">
            <XCUIElementTypeOther type="XCUIElementTypeOther">
                <XCUIElementTypeOther type="XCUIElementTypeImage">
                    <XCUIElementTypeStaticText type="XCUIElementTypeText">
                        <XCUIElementTypeStaticText type="XCUIElementTypeText">
                            <XCUIElementTypeButton type="XCUIElementTypeButton">
                                </XCUIElementTypeButton>
                            </XCUIElementTypeText>
                        </XCUIElementTypeText>
                    </XCUIElementTypeImage>
                </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)
    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">
            <XCUIElementTypeOther type="XCUIElementTypeOther">
                <XCUIElementTypeStaticText type="XCUIElementTypeText" value="米家隐私政策" />
                <XCUIElementTypeTextView type="XCUIElementTypeText" value="确定" />
            </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">
            <XCUIElementTypeOther type="XCUIElementTypeOther">
                <XCUIElementTypeOther type="XCUIElementTypeTable">
                    <XCUIElementTypeOther type="XCUIElementTypeCell">
                        <XCUIElementTypeImage type="XCUIElementTypeImage" ...>
                        <XCUIElementTypeSearchField type="XCUIElementTypeSearchField" ...>
                    </XCUIElementTypeCell>
                </XCUIElementTypeTable>
                <XCUIElementTypeButton type="XCUIElementTypeButton" ...>
                <XCUIElementTypeButton type="XCUIElementTypeButton" ...>
                <XCUIElementTypeButton type="XCUIElementTypeButton" ...>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
        <XCUIElementTypeTable type="XCUIElementTypeTable" ...>
            <XCUIElementTypeCell type="XCUIElementTypeCell" ...>
                <XCUIElementTypeImage type="XCUIElementTypeImage" ...>
                <XCUIElementTypeSearchField type="XCUIElementTypeSearchField" ...>
            </XCUIElementTypeCell>
        </XCUIElementTypeTable>
    </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"
            <XCUIElementTypeOther type="XCUIElementTypeOther"
                <XCUIElementTypeOther type="XCUIElementTypeOther"
                    <XCUIElementTypeOther type="XCUIElementTypeOther"
                        <XCUIElementTypeOther type="XCUIElementTypeImage"
                            <XCUIElementTypeImage type="XCUIElementTypeImage"
                            <XCUIElementTypeStaticText type="XCUIElementTypeText">
                                <XCUIElementTypeButton type="XCUIElementTypeButton"
                                    <XCUIElementTypeButton type="XCUIElementTypeImage"/>
                            
                        
                    
                
            
        
    """
    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 被已阅
    """
    foundAndProcessedPopup = False
    """
        必要 弹框 被已阅:
            <XCUIElementTypeOther type="XCUIElementTypeOther">
                <XCUIElementTypeImage type="XCUIElementTypeImage">
                    <XCUIElementTypeStaticText type="XCUIElementTypeText">
                        <XCUIElementTypeTextView type="XCUIElementTypeText">
                            <XCUIElementTypeButton type="XCUIElementTypeButton">
                                </XCUIElementTypeImage>
                            </XCUIElementTypeOther>
                        </XCUIElementTypeText>
                    </XCUIElementTypeImage>
                </XCUIElementTypeOther>
            </XCUIElementTypeImage>
        </XCUIElementTypeText>
    """
    hasReadChainList = [
        {
            "tag": "XCUIElementTypeOther",
            "attrs": {"enabled": "true", "visible": "true", "text": "被已阅"}
        },
        {
            "tag": "XCUIElementTypeImage",
            "attrs": {"enabled": "true", "visible": "true", "text": "被已阅"}
        },
        {
            "tag": "XCUIElementTypeButton",
            "attrs": {"enabled": "true", "visible": "true", "text": "被已阅"}
        }
    ]
    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">
        <XCUIElementTypeOther type="XCUIElementTypeOther">
            <XCUIElementTypeOther type="XCUIElementTypeOther">
                <XCUIElementTypeOther type="XCUIElementTypeOther">
                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                <XCUIElementTypeStaticText type="XCUIElementTypeText" value="取消" />
                                            </XCUIElementTypeOther>
                                        </XCUIElementTypeOther>
                                    </XCUIElementTypeOther>
                                </XCUIElementTypeOther>
                            </XCUIElementTypeOther>
                        </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", "label": "完成"}
},
]
alertGiveupSoup = CommonUtils.bsChainFind(soup, giveup)
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">
        <XCUIElementTypeOther type="XCUIElementTypeOther">
            <XCUIElementTypeButton type="XCUIElementTypeButton">
                <XCUIElementTypeStaticText type="XCUIElementTypeText">
                    </XCUIElementTypeButton>
                <XCUIElementTypeOther type="XCUIElementTypeOther">
                    <XCUIElementTypeImage type="XCUIElementTypeImage">
                        <XCUIElementTypeStaticText type="XCUIElementTypeText">
                            </XCUIElementTypeImage>
                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                            <XCUIElementTypeCollectionView type="XCUIElementTypeCollectionView">
                                <XCUIElementTypeButton type="XCUIElementTypeButton">
                                    <XCUIElementTypeStaticText type="XCUIElementTypeText">
                                        </XCUIElementTypeButton>
                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                        <XCUIElementTypeText type="XCUIElementTypeText">
                                            </XCUIElementTypeOther>
                                        <XCUIElementTypeText type="XCUIElementTypeText">
                                            </XCUIElementTypeText>
                                        <XCUIElementTypeText type="XCUIElementTypeText">
                                            </XCUIElementTypeText>
                                    </XCUIElementTypeText>
                                </XCUIElementTypeText>
                            </XCUIElementTypeText>
                        </XCUIElementTypeText>
                    </XCUIElementTypeText>
                </XCUIElementTypeText>
            </XCUIElementTypeText>
        </XCUIElementTypeText>
    </XCUIElementTypeText>
    .....
    busyRetryChainList = [
        {
            "tag": "XCUIElementTypeOther",
            "attrs": {"enabled": "true", "visible": "true", "text": "京东金融 异常页面 服务器繁忙, 请稍后重试"}
        },
        {
            "tag": "XCUIElementTypeOther",
            "attrs": {"enabled": "true", "visible": "true", "text": "京东金融 异常页面 服务器繁忙, 请稍后重试"}
        },
        {
            "tag": "XCUIElementTypeStaticText",
            "attrs": {"enabled": "true", "visible": "true", "text": "京东金融 异常页面 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 bac
            backupSoup = parentParentOtherSoup.find(
                "XCUIElementTypeButton",
                attrs={"visible": "true", "enabled": "true", "text": "京东金融 异常页面 服务器繁忙, 请稍后重试"})
            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">
        <XCUIElementTypeOther type="XCUIElementTypeOther">
            <XCUIElementTypeOther type="XCUIElementTypeImage">
                <XCUIElementTypeStaticText type="XCUIElementTypeText">
                    </XCUIElementTypeText>
                </XCUIElementTypeImage>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    </XCUIElementTypeOther>
    .....
    retryRefreshChainList = [
        {
            "tag": "XCUIElementTypeOther",
            "attrs": {"enabled": "true", "visible": "true", "name": "异常提示框"}
        },
        {
            "tag": "XCUIElementTypeOther",
            "attrs": {"enabled": "true", "visible": "true", "name": "异常提示框"}
        },
        {
            "tag": "XCUIElementTypeOther",
            "attrs": {"enabled": "true", "visible": "true", "name": "异常提示框"}
        },
    ]
    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">
            <XCUIElementTypeOther type="XCUIElementTypeOther">
                <XCUIElementTypeOther type="XCUIElementTypeOther">
                    <XCUIElementTypeOther type="XCUIElementTypeImage">
                        <XCUIElementTypeImage type="XCUIElementTypeImage">
                            <XCUIElementTypeButton type="XCUIElementTypeButton">
                                <XCUIElementTypeButton type="XCUIElementTypeButton">
                                    </XCUIElementTypeOther>
                                </XCUIElementTypeButton>
                            </XCUIElementTypeImage>
                        </XCUIElementTypeImage>
                    </XCUIElementTypeOther>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeWindow>
    """
    specialChar = "□"
    underCloseChainList = [
        {
            "tag": "XCUIElementTypeWindow",
            "attrs": self.FullScreenAttrDict,
        },
        {
            "tag": "XCUIElementTypeOther",
            "attrs": self.FullScreenAttrDict,
        },
        {
            "tag": "XCUIElementTypeButton",
            "attrs": {"enabled": "true", "visible": "true", "name": "□"},
        },
    ]
    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 smaller than
    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 首单限时福利 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 10*10 ~ 30*30
    """

    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，算作关闭按钮
        意思是：引导页往往伴随着一个普通大小的按钮，供关闭用

    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,
    .....
    foundAndProcessedPopup = False
    .....
    必要 弹框 首单限时福利 右上角 关闭按钮：
    <XCUIElementTypeWindow type="XCUIElementTypeWindow">
        ...
        <XCUIElementTypeOther type="XCUIElementTypeOther">
            ...

```

```
<XCUIElementTypeButton type="XCUIElementTypeButton">
<XCUIElementTypeImage type="XCUIElementTypeImage">
</XCUIElementTypeOther>
</XCUIElementTypeWindow>

必要 我的 蒙层弹框 首次使用引导页:
<XCUIElementTypeWindow type="XCUIElementTypeWindow">
    ...
    <XCUIElementTypeOther type="XCUIElementTypeOther">
        <XCUIElementTypeOther type="XCUIElementTypeOther">
            <XCUIElementTypeOther type="XCUIElementTypeOther">
                <XCUIElementTypeOther type="XCUIElementTypeImage">
                    <XCUIElementTypeImage type="XCUIElementTypeImage">
                    <XCUIElementTypeButton type="XCUIElementTypeButton">
                        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
                            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
                                <XCUIElementTypeButton type="XCUIElementTypeButton">
                                    </XCUIElementTypeOther>
                                </XCUIElementTypeOther>
                            </XCUIElementTypeStaticText>
                        </XCUIElementTypeButton>
                    </XCUIElementTypeImage>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    </XCUIElementTypeWindow>

必要 弹框 推荐开通以下授权 关闭按钮在弹框下面:
<XCUIElementTypeApplication type="XCUIElementTypeApplication">
    <XCUIElementTypeWindow type="XCUIElementTypeWindow">
        ...
        <XCUIElementTypeOther type="XCUIElementTypeOther">
            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
                <XCUIElementTypeOther type="XCUIElementTypeOther">
                    <XCUIElementTypeOther type="XCUIElementTypeImage">
                        <XCUIElementTypeImage type="XCUIElementTypeImage">
                        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
                            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
                                </XCUIElementTypeOther>
                            </XCUIElementTypeOther>
                        </XCUIElementTypeStaticText>
                    </XCUIElementTypeImage>
                </XCUIElementTypeOther>
            </XCUIElementTypeStaticText>
        </XCUIElementTypeOther>
        <XCUIElementTypeButton type="XCUIElementTypeButton">
            </XCUIElementTypeOther>
        </XCUIElementTypeWindow>
        ...
        # noNameP = re.compile("???")
        noNameP = False
        possibleCloseChainList = [
            {
                "tag": "XCUIElementTypeWindow",
                "attrs": {"enabled": "true", "visible": "true", "name": "蒙层弹框首次使用引导页"}
            },
            {
                "tag": "XCUIElementTypeOther",
                "attrs": {"enabled": "true", "visible": "true", "name": "蒙层弹框首次使用引导页"}
            },
            {
                "tag": "XCUIElementTypeButton",
                "attrs": {"enabled": "true", "visible": "true", "name": "蒙层弹框首次使用引导页"}]
```

```

        "attrs": {"enabled": "true", "visible": "true", '},
    ],
possibleCloseSoup = CommonUtils.bsChainFind(soup, poss:

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"
        },
        {
            "tag": "XCUIElementTypeWindow",
            "attrs": {"enabled": "true", "visible": "true"
        },
        {
            "tag": "XCUIElementTypeButton",
            "attrs": {"enabled": "true", "visible": "true"
        },
    ]
possibleCloseSoup = CommonUtils.bsChainFind(soup, (
if possibleCloseSoup:
    isCloseUnderPopup = True

if possibleCloseSoup:
    # check is match close button rule or not
    isValidSize = False
    isAlmostSameWH = False
    # isNexSiblingImage = False
    # isImageSizeValid = False
    isPossibleClose = True

    if isPossibleClose:
        soupAttrDict = possibleCloseSoup.attrs # {'enat
        logging.debug("possibleCloseSoup 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 <= Button
        isValidHeight = ButtonMinWH <= height <= Button
        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
    # nextSiblingeSoupList = possibleCloseSoup.next_siblings
    nextSiblingeSoupGenerator = possibleCloseSoup.next_siblings
    nextSiblingeSoupList = list(nextSiblingeSoupGenerator)

    hasLargeImage = CommonUtils.isContainSpecificSize(nextSiblingeSoupList)
    isPossibleClose = hasLargeImage

if not isPossibleClose:
    hasNormalButton = CommonUtils.isContainSpecificSize(nextSiblingeSoupList)
    isPossibleClose = hasNormalButton

if not isPossibleClose:
    if isCloseUnderPopup:
        # prevSiblingSoupGenerator = possibleCloseSoup.previous_sibling
        # prevPrevSiblingSoupList = list(prevSiblingSoupGenerator)
        # hasLargeOther = CommonUtils.isContainSpecificSize(nextSiblingeSoup)
        prevSiblingSoup = possibleCloseSoup.previous_sibling
        if prevSiblingSoup:
            prevPrevSiblingSoup = prevSiblingSoup.previous_sibling
            prevPrevSiblingSoupList = [prevPrevSiblingSoup]
            hasLargeOther = CommonUtils.isContainSpecificSize(nextSiblingeSoup)
            isPossibleClose = hasLargeOther

    if isPossibleClose:
        foundAndProcessedPopup = self.findAndClickButton(nextSiblingeSoupList)

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">
            <XCUIElementTypeOther type="XCUIElementTypeOther">
                <XCUIElementTypeStaticText type="XCUIElementTypeText" value="关闭" />
                <XCUIElementTypeButton type="XCUIElementTypeButton" value="关闭" />
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>

    京东金融 tab2财富 弹框 我是Max alertClose:
        <XCUIElementTypeOther type="XCUIElementTypeOther">
            ...
            <XCUIElementTypeOther type="XCUIElementTypeOther">
                <XCUIElementTypeButton type="XCUIElementTypeButton" value="关闭" />
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>

    必要 弹框 右上角关闭按钮 name中有close:
        <XCUIElementTypeOther type="XCUIElementTypeOther">
            <XCUIElementTypeOther type="XCUIElementTypeOther">
                <XCUIElementTypeOther type="XCUIElementTypeOther">
                    <XCUIElementTypeButton type="XCUIElementTypeButton" value="关闭" />
                    <XCUIElementTypeImage type="XCUIElementTypeImage" value="关闭" />
                    <XCUIElementTypeStaticText type="XCUIElementTypeText" value="关闭" />
                    <XCUIElementTypeOther type="XCUIElementTypeOther" value="关闭" />
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>

    必要 验证码登录页 左上角 关闭 按钮 name中含close:
        <XCUIElementTypeOther type="XCUIElementTypeOther">
            <XCUIElementTypeOther type="XCUIElementTypeOther">
                <XCUIElementTypeOther type="XCUIElementTypeOther">
                    <XCUIElementTypeButton type="XCUIElementTypeButton" value="关闭" />
                    <XCUIElementTypeImage type="XCUIElementTypeImage" value="关闭" />
                    <XCUIElementTypeStaticText type="XCUIElementTypeText" value="关闭" />
                    <XCUIElementTypeOther type="XCUIElementTypeOther" value="关闭" />
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
        .....
        commonCloseP = re.compile("close", flags=re.I)
        # com icon black close u
        # jr fianncing alertClose@3x
        # rights close white
        # login close
        commonCloseChainList = [
            {
                "tag": "XCUIElementTypeOther",
                "attrs": {"enabled": "true", "visible": "true", "value": "关闭"}
            },
            {

```

```
        "tag": "XCUIElementTypeOther",
        # "attrs": {"enabled":"true", "visible":"true"},
        # "attrs": {"enabled":"true", "visible":"true"}
        "attrs": {"enabled":"true", "visible":"true"}
    },
{
    "tag": "XCUIElementTypeButton",
    "attrs": {"enabled":"true", "visible":"true", "label": "Close", "name": "Close", "type": "button", "value": "Close", "color": "#000000", "textColor": "#FFFFFF", "fontStyle": "bold", "fontSize": 17, "fontFamily": "System", "borderWidth": 1, "borderColor": "#000000", "borderRadius": 0, "padding": 10, "width": 44, "height": 44, "centerPosition": true}
},
]
commonCloseSoup = CommonUtils.bsChainFind(soup, commonClose)
if commonCloseSoup:
    # self.clickElementCenterPosition(commonCloseSoup)
    # foundAndProcessedPopup = True

    # so change to wda query element then click
    foundAndProcessedPopup = self.findAndClickButtonElement(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">
#             ...
#             <XCUIElementTypeOther type="XCUIElementTypeOther">
#                 <XCUIElementTypeButton type="XCUIElementTypeButton">
#             ...
#         alertCloseP = re.compile("alertClose")
#         alertCloseChainList = [
#             {
#                 "tag": "XCUIElementTypeOther",
#                 "attrs": {"enabled": "true", "visible": "true"},
#             },
#             {
#                 "tag": "XCUIElementTypeOther",
#                 "attrs": {"enabled": "true", "visible": "true"},
#             },
#             {
#                 "tag": "XCUIElementTypeButton",
#                 "attrs": {"enabled": "true", "visible": "true"},
#             },
#         ]
#         alertCloseSoup = CommonUtils.bsChainFind(soup, alertCloseP)
#         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">
#             <XCUIElementTypeOther type="XCUIElementTypeOther">
#                 <XCUIElementTypeStaticText type="XCUIElementTypeText">
#                     <XCUIElementTypeButton type="XCUIElementTypeButton">
#                 </XCUIElementTypeButton>
#             ...
#         containCloseP = re.compile("close") # com icon black
#         closeChainList = [
#             {
#                 "tag": "XCUIElementTypeOther",
#                 "attrs": {"enabled": "true", "visible": "true"},
#             },
#             {
#                 "tag": "XCUIElementTypeOther",
#                 "attrs": {"enabled": "true", "visible": "true"},
#             },
#         ]
```

```
#     {
#         "tag": "XCUIElementTypeButton",
#         "attrs": {"enabled":"true", "visible":"true"},
#     },
# ]
# closeSoup = CommonUtils.bsChainFind(soup, closeChain)
# 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">
        <XCUIElementTypeOther type="XCUIElementTypeOther">
            <XCUIElementTypeOther type="XCUIElementTypeOther">
                <XCUIElementTypeStaticText type="XCUIElementTypeText">
                    <XCUIElementTypeButton type="XCUIElementTypeButton">
                </XCUIElementTypeButton>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    .....
    jumpOverChainList = [
        {
            "tag": "XCUIElementTypeOther",
            "attrs": {"enabled":"true", "visible":"true", "name": "京东金融", "text": "京东金融"}, "text": "京东金融"
        },
        {
            "tag": "XCUIElementTypeOther",
            "attrs": {"enabled":"true", "visible":"true", "name": "京东金融", "text": "京东金融"}, "text": "京东金融"
        },
        {
            "tag": "XCUIElementTypeButton",
            "attrs": {"enabled":"true", "visible":"true", "name": "京东金融", "text": "京东金融"}, "text": "京东金融"
        },
    ]
    jumpOverSoup = CommonUtils.bsChainFind(soup, jumpOverChainList)
    if jumpOverSoup:
        self.clickElementCenterPosition(jumpOverSoup)
        foundAndProcessedPopup = True
    return foundAndProcessedPopup
```

iOSProcessAlertUseWirelessData


```
        </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", "label": "OK"}
        },
    ]
    wifiCellularSoup = CommonUtils.bsChainFind(soup, wifiCellularChainList)
    if wifiCellularSoup:
        # found 无线局域网与蜂窝移动网络 but
        # self.clickElementCenterPosition(wifiCellularSoup)
        # foundAndProcessedPopup = True

        # # change to wda query element then click by element
        # curName = wifiCellularSoup.attrs["name"] # 好
        # wifiCellularButtonQuery = {"type": "XCUIElementTypeButton", "label": "OK"}
        # foundAndClicked = self.findAndClickElement(wifiCellularButtonQuery)
        # foundAndProcessedPopup = foundAndClicked
        foundAndProcessedPopup = self.findAndClickButtonElement(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">
            <XCUIElementTypeOther type="XCUIElementTypeOther">
                <XCUIElementTypeOther type="XCUIElementTypeOther">
                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
................................................................
accessYourP = re.compile("想访问您的") # “恒易贷”想访问您的
```

```
okChainList = [
    {
        "tag": "XCUIElementTypeAlert",
        "attrs": {"enabled": "true", "visible": "true", "name": "OK", "type": "XCUIElementTypeButton"}
    },
    {
        "tag": "XCUIElementTypeOther",
        "attrs": {"enabled": "true", "visible": "true", "type": "XCUIElementTypeImage", "name": "cancel", "label_inset": "left", "label_text": "Cancel", "label_type": "XCUIElementTypeText", "label_x": 10, "label_y": 10}
    },
    {
        "tag": "XCUIElementTypeButton",
        "attrs": {"enabled": "true", "visible": "true", "type": "XCUIElementTypeText", "label_inset": "center", "label_text": "OK", "label_type": "XCUIElementTypeText", "label_x": 50, "label_y": 50}
    }
]
okSoup = CommonUtils.bsChainFind(soup, okChainList)
if okSoup:
    # clickCenterPosition(curSession, okSoup.attrs)
    # foundAndProcessedPopup = True

    # # Note: seems actual click is No allow button? =>
    # # change to wda query element then click for make it work
    # curName = okSoup.attrs["name"] # 好
    # okButtonQuery = {"type": "XCUIElementTypeButton", "name": curName}
    # foundAndClicked = self.findAndClickElement(okButtonQuery)
    # foundAndProcessedPopup = foundAndClicked
    foundAndProcessedPopup = self.findAndClickButtonElement(okButtonQuery)

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">
        <XCUIElementTypeOther type="XCUIElementTypeText" value="京东金融
            想给您发送信息 允许:<br/>" />
        <XCUIElementTypeOther type="XCUIElementTypeImage" />
        <XCUIElementTypeOther type="XCUIElementTypeImage" />
        <XCUIElementTypeOther type="XCUIElementTypeImage" />
        <XCUIElementTypeOther type="XCUIElementTypeImage" />
        <XCUIElementTypeImage />
        <XCUIElementTypeText value="取消" />
        <XCUIElementTypeText value="允许" />
    </XCUIElementTypeAlert>
    <XCUIElementTypeText value="京东金融" />
    <XCUIElementTypeText value="弹框 允许“京东金融”在您使用该应用时访问您的位置吗?" />
    <XCUIElementTypeAlert type="XCUIElementTypeAlert">
        <XCUIElementTypeOther type="XCUIElementTypeText" value="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?<br/>" />
        <XCUIElementTypeOther type="XCUIElementTypeImage" />
```

```
<XCUIElementTypeOther type="XCUIElementTypeAlert">
    <XCUIElementTypeOther type="XCUIElementTypeAlert">
        <XCUIElementTypeOther type="XCUIElementTypeAlert">
            <XCUIElementTypeOther type="XCUIElementTypeAlert">
                <XCUIElementTypeOther type="XCUIElementTypeAlert">
                    <XCUIElementTypeOther type="XCUIElementTypeAlert">
                        <XCUIElementTypeOther type="XCUIElementTypeAlert">
                            <XCUIElementTypeOther type="XCUIElementTypeAlert">
                                <XCUIElementTypeOther type="XCUIElementTypeAlert">
                                    <XCUIElementTypeOther type="XCUIElementTypeAlert">
                                        <XCUIElementTypeOther type="XCUIElementTypeAlert">
                                            <XCUIElementTypeOther type="XCUIElementTypeAlert">
                                                <XCUIElementTypeOther type="XCUIElementTypeAlert">
                                                    <XCUIElementTypeOther type="XCUIElementTypeAlert">
                                                        <XCUIElementTypeOther type="XCUIElementTypeAlert">
                                                            <XCUIElementTypeOther type="XCUIElementTypeAlert">
                                                                <XCUIElementTypeOther type="XCUIElementTypeAlert">
                                                                    <XCUIElementTypeOther type="XCUIElementTypeAlert">
                                                                        <XCUIElementTypeOther type="XCUIElementTypeAlert">
                                                                            <XCUIElementTypeOther type="XCUIElementTypeAlert">
                                                                                <XCUIElementTypeOther type="XCUIElementTypeAlert">
                                                                                    <XCUIElementTypeOther type="XCUIElementTypeAlert">
                                                                                        <XCUIElementTypeOther type="XCUIElementTypeAlert">
                                                                                            <XCUIElementTypeOther type="XCUIElementTypeAlert">
                                                                                                <XCUIElementTypeOther type="XCUIElementTypeAlert">
                                                                                                    <XCUIElementTypeOther type="XCUIElementTypeAlert">
                                                                                                        <XCUIElementTypeOther type="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

```
<XCUIElementTypeAlert type="XCUIElementTypeAlert">
    <XCUIElementTypeOther type="XCUIElementTypeAlert">
        <XCUIElementTypeOther type="XCUIElementTypeAlert">
            <XCUIElementTypeOther type="XCUIElementTypeAlert">
                <XCUIElementTypeOther type="XCUIElementTypeAlert">
                    <XCUIElementTypeOther type="XCUIElementTypeAlert">
                        <XCUIElementTypeOther type="XCUIElementTypeAlert">
                            <XCUIElementTypeOther type="XCUIElementTypeAlert">
                                <XCUIElementTypeOther type="XCUIElementTypeAlert">
                                    <XCUIElementTypeOther type="XCUIElementTypeAlert">
................................................................
```

恒易贷 想给您发送通知 允许:

```

                <XCUIElementTypeOther type="XCUIElementTypeAlert">
                </XCUIElementTypeOther>
                <XCUIElementTypeOther type="XCUIElementTypeOther">
                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                 ...

```

....

```

# allowP = re.compile("(始终)?允许") # will also match "
allowP = re.compile("^(始终)?允许$") # only match "允许"
allowChainList = [
    {
        "tag": "XCUIElementTypeAlert",
        "attrs": {"enabled": "true", "visible": "true"}
    },
    {
        "tag": "XCUIElementTypeOther",
        "attrs": {"enabled": "true", "visible": "true"}
    },
    {
        "tag": "XCUIElementTypeButton",
        "# attrs": {"enabled": "true", "visible": "true",
        "attrs": {"enabled": "true", "visible": "true", "name": "允许"}}
    },
]
allowSoup = CommonUtils.bsChainFind(soup, allowChainList)
if allowSoup:
    # clickCenterPosition(curSession, allowSoup.attrs)
    # foundAndProcessedPopup = True

    # # above click position not work for 允许 -> actual
    # # change to use wda to find 允许 then click element
    # curName = allowSoup.attrs["name"] # 允许 / 始终允许
    # # allowButtonQuery = {"type": "XCUIElementTypeButton", "name": "允许"}
    # # allowButtonQuery = {"type": "XCUIElementTypeButton", "name": "始终允许"}
    # foundAndClickedAllow = self.findAndClickElement(a

```

```
# foundAndProcessedPopup = foundAndClickeAllow
foundAndProcessedPopup = self.findAndClickButtonElement()
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 我接受:
    <XCUIElementTypeWindow type="XCUIElementTypeWindow">
        <XCUIElementTypeOther type="XCUIElementTypeOther">
            <XCUIElementTypeOther type="XCUIElementTypeOther">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
                        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                <XCUIElementTypeButton type="XCUIElementTypeButton">
                                    <XCUIElementTypeButton type="XCUIElementTypeButton">
                                        </XCUIElementTypeOther>
                                    </XCUIElementTypeButton>
                                </XCUIElementTypeWindow>
                            .....
                            userPrivacyIAcceptChainList = [
                                {
                                    "tag": "XCUIElementTypeWindow",
                                    "attrs": {"enabled": "true", "visible": "true", "name": "恒易贷 用户意愿保护协议 我接受:"},
                                },
                                {
                                    "tag": "XCUIElementTypeOther",
                                    "attrs": {"enabled": "true", "visible": "true", "name": "恒易贷 用户意愿保护协议 我接受:"},
                                },
                                {
                                    "tag": "XCUIElementTypeButton",
                                    "attrs": {"enabled": "true", "visible": "true", "name": "恒易贷 用户意愿保护协议 我接受:"},
                                },
                            ]
                            userPrivacyIAcceptSoup = CommonUtils.bsChainFind(soup, userPrivacyIAcceptChainList)
                            if userPrivacyIAcceptSoup:
                                self.clickElementCenterPosition(userPrivacyIAcceptSoup)
                                foundAndProcessedPopup = True
    return foundAndProcessedPopup
```

iOS Process Common User Agreement Agree e

```
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">
            ...
            <XCUIElementTypeOther type="XCUIElementTypeOther">
                <XCUIElementTypeOther type="XCUIElementTypeOther">
                    <XCUIElementTypeButton type="XCUIElementTypeButton">
                        <XCUIElementTypeButton type="XCUIElementTypeButton">
                            </XCUIElementTypeButton>
                        </XCUIElementTypeButton>
                    </XCUIElementTypeButton>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    """

    必要 用户服务协议及必要隐私政策 同意:
    <XCUIElementTypeOther type="XCUIElementTypeOther">
        <XCUIElementTypeOther type="XCUIElementTypeOther">
            <XCUIElementTypeStaticText type="XCUIElementTypeText">
                <XCUIElementTypeTextView type="XCUIElementTypeText">
                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                            <XCUIElementTypeButton type="XCUIElementTypeButton">
                                <XCUIElementTypeButton type="XCUIElementTypeButton">
                                    <XCUIElementTypeButton type="XCUIElementTypeButton">
                                        </XCUIElementTypeButton>
                                    </XCUIElementTypeButton>
                                </XCUIElementTypeButton>
                            </XCUIElementTypeOther>
                        </XCUIElementTypeOther>
                    </XCUIElementTypeText>
                </XCUIElementTypeText>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    </XCUIElementTypeOther>

    斑马AI课 弹框 个人信息保护政策 同意:
    <XCUIElementTypeWindow type="XCUIElementTypeWindow">
        <XCUIElementTypeOther type="XCUIElementTypeOther">
            <XCUIElementTypeOther type="XCUIElementTypeOther">
                <XCUIElementTypeOther type="XCUIElementTypeOther">
                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                        <XCUIElementTypeStaticText type="XCUIElementTypeText">
                            <XCUIElementTypeTextView type="XCUIElementTypeText">
                                <XCUIElementTypeLink type="XCUIElementTypeText">
                                    <XCUIElementTypeLink type="XCUIElementTypeText">
                                        <XCUIElementTypeLink type="XCUIElementTypeText">
                                            </XCUIElementTypeText>
                                        </XCUIElementTypeText>
                                    </XCUIElementTypeText>
                                </XCUIElementTypeText>
                            </XCUIElementTypeText>
                        </XCUIElementTypeOther>
                    </XCUIElementTypeOther>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    </XCUIElementTypeWindow>
    """
    commonAgreeChainList = [
        {
            "tag": "XCUIElementTypeOther",
            "attrs": {"enabled": "true", "visible": "true", "name": "同意", "type": "button", "text": "同意", "value": "1"}, "value": "1"
        }
    ]
```

```
        },
        {
            "tag": "XCUIElementTypeOther",
            # "attrs": {"enabled":"true", "visible":"true"},
            "attrs": {"enabled":"true", "visible":"true"}
        },
        {
            "tag": "XCUIElementTypeButton",
            "attrs": {"enabled":"true", "visible":"true", "type": "button"}
        },
    ]
commonAgreeSoup = CommonUtils.bsChainFind(soup, commonAgreeXpath)
if commonAgreeSoup:
    self.clickElementCenterPosition(commonAgreeSoup)
    foundAndProcessedAgreement = True
return foundAndProcessedAgreement
```

iOSProcessPopupKnow

```
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">
            <XCUIElementTypeOther type="XCUIElementTypeOther">
                <XCUIElementTypeScrollView type="XCUIElementTypeOther">
                    <XCUIElementTypeStaticText type="XCUIElementTypeText">
                        <XCUIElementTypeOther type="XCUIElementTypeText">
                    </XCUIElementTypeText>
                </XCUIElementTypeScrollView>
                <XCUIElementTypeOther type="XCUIElementTypeOther">
                    <XCUIElementTypeButton type="XCUIElementTypeText">
                </XCUIElementTypeText>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    """

    恒易贷 权限获取说明 我知道了:
    <XCUIElementTypeOther type="XCUIElementTypeOther">
        <XCUIElementTypeOther type="XCUIElementTypeOther">
            <XCUIElementTypeOther type="XCUIElementTypeTable">
                <XCUIElementTypeStaticText type="XCUIElementTypeText">
                    <XCUIElementTypeStaticText type="XCUIElementTypeText">
                        <XCUIElementTypeTable type="XCUIElementTypeTable">
                            <XCUIElementTypeOther type="XCUIElementTypeText">
                                <XCUIElementTypeCell type="XCUIElementTypeText">
                                    <XCUIElementTypeImage type="XCUIElementTypeImage">
                                    <XCUIElementTypeStaticText type="XCUIElementTypeText">
                                    <XCUIElementTypeStaticText type="XCUIElementTypeText">
                            </XCUIElementTypeText>
                        </XCUIElementTypeTable>
                    <XCUIElementTypeCell type="XCUIElementTypeText">
                        <XCUIElementTypeImage type="XCUIElementTypeImage">
                        <XCUIElementTypeStaticText type="XCUIElementTypeText">
                        <XCUIElementTypeStaticText type="XCUIElementTypeText">
                    </XCUIElementTypeText>
                </XCUIElementTypeTable>
            <XCUIElementTypeCell type="XCUIElementTypeText">
                <XCUIElementTypeImage type="XCUIElementTypeImage">
                <XCUIElementTypeStaticText type="XCUIElementTypeText">
                <XCUIElementTypeStaticText type="XCUIElementTypeText">
            </XCUIElementTypeText>
        </XCUIElementTypeTable>
        <XCUIElementTypeText type="XCUIElementTypeText">
            <XCUIElementTypeImage type="XCUIElementTypeImage">
            <XCUIElementTypeStaticText type="XCUIElementTypeText">
            <XCUIElementTypeStaticText type="XCUIElementTypeText">
        </XCUIElementTypeText>
    </XCUIElementTypeOther>
    <XCUIElementTypeText type="XCUIElementTypeText">
        <XCUIElementTypeImage type="XCUIElementTypeImage">
        <XCUIElementTypeText type="XCUIElementTypeText">
        <XCUIElementTypeText type="XCUIElementTypeText">
    </XCUIElementTypeText>
</XCUIElementTypeOther>
```

```
</XCUIElementTypeOther>
.....
iKnowChainList = [
    {
        "tag": "XCUIElementTypeOther",
        "attrs": {"enabled":"true", "visible":"true", "name": "Call Now", "label": "Call Now", "type": "button", "value": null}
    },
    {
        "tag": "XCUIElementTypeOther",
        "attrs": {"enabled":"true", "visible":"true", "name": "Call Now", "label": "Call Now", "type": "button", "value": null}
    },
    {
        "tag": "XCUIElementTypeButton",
        "attrs": {"enabled":"true", "visible":"true", "name": "Call Now", "label": "Call Now", "type": "button", "value": null}
    }
]
iKnowSoup = CommonUtils.bsChainFind(soup, iKnowChainList)
if iKnowSoup:
    self.clickElementCenterPosition(iKnowSoup)
    foundAndProcessedPopup = True
return foundAndProcessedPopup
```

iOSProcessPopupPhoneCall

```
def iOSProcessPopupPhoneCall(self, soup):
    """iOS 通用弹框 呼叫 -> 点击 取消"""
    foundAndProcessedPopup = False
    .....
    iOS 通用弹框 呼叫 打电话:
        <XCUIElementTypeAlert type="XCUIElementTypeAlert">
            <XCUIElementTypeOther type="XCUIElementTypeOther">
                <XCUIElementTypeOther type="XCUIElementTypeOther">
                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                    <XCUIElementTypeOther type="XCUIElementTypeStaticText">
                                                        </XCUIElementTypeOther>
                                                    </XCUIElementTypeOther>
                                                </XCUIElementTypeOther type="XCUIElementTypeOther">
                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                 <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                     <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                         <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                             <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                                 <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                                 <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                                 <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                                 <XCUIElementTypeOther type="XCUIElementTypeOther">
................................................................
callChainList = [
{
    "tag": "XCUIElementTypeAlert",
    "attrs": {"enabled": "true", "visible": "true"}  
.....
```

```
        },
        {
            "tag": "XCUIElementTypeOther",
            "attrs": {"enabled":"true", "visible":"true"}
        },
        {
            "tag": "XCUIElementTypeButton",
            "attrs": {"enabled":"true", "visible":"true", "label": "取消"}
        }
    ]
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"})
    #         if cancelSoup:
    #             clickCenterPosition(curSession, cancelSoup)
    #             foundAndProcessedPopup = True

    # # above click position not work for 取消 !!!
    # # change to find 取消 then click element
    # cancelButtonQuery = {"type":"XCUIElementTypeButton", "label": "取消"}
    # foundAndClickeCancel = self.findAndClickElement(cancelButtonQuery)
    # foundAndProcessedPopup = foundAndClickeCancel
    foundAndProcessedPopup = self.findAndClickButtonElement("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
    enabled="true" visible="true" x="0" y="0" width="4"
    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
    <XCUIElementTypeOther type="XCUIElementTypeOther">
        <XCUIElementTypeOther type="XCUIElementTypeOther">
            <XCUIElementTypeOther type="XCUIElementTypeOther">
                <XCUIElementTypeOther type="XCUIElementTypeOther">
                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                
    .....
    allTopOtherSoupList = soup.find_all(
        "XCUIElementTypeOther",
        attrs={"enabled":"true", "visible":"true", "x":"0"
    )
    for eachTopOtherSoup in allTopOtherSoupList:
        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 + (possibleCloseButtonWidth / 2)
            # possibleCloseButtonCenterY = possibleCloseButtonY0 + (possibleCloseButtonHeight / 2)
            possibleCloseButtonCenterX = int((possibleCloseButtonX0 + possibleCloseButtonX1) / 2)
            possibleCloseButtonCenterY = int((possibleCloseButtonY0 + possibleCloseButtonY1) / 2)
            possibleCloseButtonCenterPositon = (possibleCloseButtonCenterX, possibleCloseButtonCenterY)
            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:
        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
    FullScreenSizeLevel = 3 # special: 无忧筹 点击 专属老师 后的

    FullScreenAttr = {"type": "XCUIElementTypeOther", "enabled": "true", "visible": "true", "x": "0", "y": "0", "level": FullScreenSizeLevel}

    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  
isPossiblePopup = curFullScreenOtherLevel == FullScreenSizeMinLevel  
if isPossiblePopup:  
    curChildOtherList = curSoup.find_all(  
        "XCUIElementTypeOther",  
        attrs={"type": "XCUIElementTypeOther", "enabled": "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">
        <XCUIElementTypeOther type="XCUIElementTypeOther">
            <XCUIElementTypeOther type="XCUIElementTypeOther">
                <XCUIElementTypeOther type="XCUIElementTypeOther">
                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                 .....
```

```
photoCameraChainList = [
```

```
{
```

```

        "tag": "XCUIElementTypeOther",
        "attrs": {"enabled":"true", "visible":"true"}
    },
    {
        "tag": "XCUIElementTypeTable",
        "attrs": {"enabled":"true", "visible":"true", "isGrouped": "true", "isSectionHeader": "false", "rowCount": "1", "rowType": "SectionHeader", "sectionHeaderLevel": "1", "text": "Photo Camera", "value": "Photo Camera"}, "children": [
            {
                "tag": "XCUIElementTypeStaticText",
                "attrs": {"enabled":"true", "visible":"true", "text": "Photo camera", "value": "Photo camera"}, "children": []
            }
        ]
    }
}
photoCameraSoup = CommonUtils.bsChainFind(soup, photoCamera)
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["tag"]
                    if curType == "XCUIElementTypeOther":
                        # next sibling's first XCUIElement
                        nextOtherSoup = eachNextSibling
                        break

            if nextOtherSoup:
                cancelSoup = nextOtherSoup.find(
                    "XCUIElementTypeButton",
                    attrs={"visible":"true", "enabled":"true", "label": "Cancel", "value": "Cancel"}, )
                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">
        <XCUIElementTypeOther type="XCUIElementTypeOther">
            <XCUIElementTypeOther type="XCUIElementTypeOther">
                <XCUIElementTypeOther type="XCUIElementTypeOther">
                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                            <XCUIElementTypeOther type="XCUIElementTypeStatic" />
                                            <XCUIElementTypeStatic />
                                        </XCUIElementTypeOther>
                                    </XCUIElementTypeOther>
                                </XCUIElementTypeOther>
                            </XCUIElementTypeOther>
                        </XCUIElementTypeOther>
                    </XCUIElementTypeOther>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    </XCUIElementTypeAlert>
    .....
    康爱公社 弹框 提示 请登录后再进行操作：
    <XCUIElementTypeAlert type="XCUIElementTypeAlert">
        <XCUIElementTypeOther type="XCUIElementTypeOther">
            <XCUIElementTypeOther type="XCUIElementTypeOther">
                <XCUIElementTypeOther type="XCUIElementTypeOther">
                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                 <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                     <XCUIElementTypeOther type="XCUIElementTypeOther">
                                         </XCUIElementTypeOther>
                                     </XCUIElementTypeOther>
                                 </XCUIElementTypeOther>
                               </XCUIElementTypeOther>
                             </XCUIElementTypeOther>
                           </XCUIElementTypeOther>
                         </XCUIElementTypeOther>
                       </XCUIElementTypeOther>
                     </XCUIElementTypeOther>
                   </XCUIElementTypeOther>
                 </XCUIElementTypeOther>
               </XCUIElementTypeOther>
             </XCUIElementTypeOther>
           </XCUIElementTypeOther>
         </XCUIElementTypeOther>
       </XCUIElementTypeOther>
     </XCUIElementTypeAlert>
```

```
</XCUIElementTypeOther>
<XCUIElementTypeOther type="XCUIElementTypeAlert">
    <XCUIElementTypeOther type="XCUIElementTypeOther">
        <XCUIElementTypeOther type="XCUIElementTypeStaticText">
            <XCUIElementTypeStaticText>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther">
            <XCUIElementTypeOther type="XCUIElementTypeOther">
                <XCUIElementTypeOther type="XCUIElementTypeText">
                    <XCUIElementTypeText>
                        <XCUIElementTypeText>
                            <XCUIElementTypeText>
                                <XCUIElementTypeText>
                                    <XCUIElementTypeText>
                                        <XCUIElementTypeText>
                                            <XCUIElementTypeText>
                                                <XCUIElementTypeText>
                                                    <XCUIElementTypeText>
                                                        <XCUIElementTypeText>
                                                            <XCUIElementTypeText>
                                                                <XCUIElementTypeText>
                                                                    <XCUIElementTypeText>
                                                                        <XCUIElementTypeText>
                                                                            <XCUIElementTypeText>
                                                                                <XCUIElementTypeText>
                                                                                    <XCUIElementTypeText>
                                                                                        <XCUIElementTypeText>
                                                                                            <XCUIElementTypeText>
                                                                                                <XCUIElementTypeText>
                                                                                                 <XCUIElementTypeText>
                                                                                                     <XCUIElementTypeText>
                                                                                                         <XCUIElementTypeText>
                                                                                                             <XCUIElementTypeText>
                                                                                                                 <XCUIElementTypeText>
                                                                                                                 <XCUIElementTypeText>
                                                                                                                 <XCUIElementTypeText>
................................................................
noteTipP = re.compile("((提醒)|(提示))")
alertConfirmChainList = [
    {
        "tag": "XCUIElementTypeAlert",
        # "attrs": {"enabled":"true", "visible":"true"}
        "attrs": {"enabled":"true", "visible":"true", "text": "确定"},

    },
    {
        "tag": "XCUIElementTypeOther",
        "attrs": {"enabled":"true", "visible":"true"}}, { "tag": "XCUIElementTypeButton", "attrs": {"enabled":"true", "visible":"true", "text": "取消"}}, ]
]
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">
        <XCUIElementTypeOther type="XCUIElementTypeOther">
            <XCUIElementTypeOther type="XCUIElementTypeOther">
                <XCUIElementTypeAlert type="XCUIElementTypeAlert">
                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
................................................................
    alertCancelChainList = [
```

```
{
    "tag": "XCUIElementTypeAlert",
    "attrs": {"enabled":"true", "visible":"true"}
},
{
    "tag": "XCUIElementTypeOther",
    "attrs": {"enabled":"true", "visible":"true"}
},
{
    "tag": "XCUIElementTypeButton",
    "attrs": {"enabled":"true", "visible":"true", "label": "取消", "type": "button", "name": "cancel", "value": "cancel", "color": "#000000", "textColor": "#FFFFFF", "fontStyle": "bold", "fontSize": 17, "fontFamily": "PingFangSC-Medium", "width": 100, "height": 44, "x": 150, "y": 150}
},
]
alertCancelSoup = CommonUtils.bsChainFind(soup, alertCancel)
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">
        ...
        <XCUIElementTypeOther type="XCUIElementTypeOther">
            <XCUIElementTypeStaticText type="XCUIElementTypeText" value="您还未登录" />
        </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther">
            <XCUIElementTypeStaticText type="XCUIElementTypeText" value="康爱公社" />
        </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther">
            <XCUIElementTypeStaticText type="XCUIElementTypeText" value="登录" />
        </XCUIElementTypeOther>
        <XCUIElementTypeOther type="XCUIElementTypeOther">
            <XCUIElementTypeStaticText type="XCUIElementTypeText" value="取消" />
        </XCUIElementTypeOther>
    </XCUIElementTypeOther>
    .....
    stillNotLoginChainList = [
        {
            "tag": "XCUIElementTypeOther",
            "attrs": {"enabled": "true", "x": "0", "width": "100%"},
        },
        {
            "tag": "XCUIElementTypeOther",
            "attrs": {"visible": "true", "enabled": "true"}
        },
        {
            "tag": "XCUIElementTypeStaticText",
            "attrs": {"visible": "true", "enabled": "true", "color": "#000000", "fontStyle": "normal", "fontWeight": "bold", "text": "您还未登录", "size": "14", "textColor": "#000000", "textType": "copyable", "type": "XCUIElementTypeText", "value": "您还未登录"}, "text": "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", "text": "您还未登录"})
            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"
        <XCUIElementTypeOther type="XCUIElementTypeOther"
            <XCUIElementTypeOther type="XCUIElementTypeOther"
                <XCUIElementTypeOther type="XCUIElementTypeOther"
                    <XCUIElementTypeOther type="XCUIElementTypeOther"
                        <XCUIElementTypeOther type="XCUIElementTypeOther"
                            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
                                </XCUIElementTypeOther>
                            <XCUIElementTypeOther type="XCUIElementTypeOther"
                                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
                                    </XCUIElementTypeOther>
                                <XCUIElementTypeOther type="XCUIElementTypeOther"
                                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
                                        </XCUIElementTypeOther>
                                    <XCUIElementTypeButton type="XCUIElementTypeButton"/>
                                <XCUIElementTypeOther type="XCUIElementTypeOther"
                                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
                                        </XCUIElementTypeOther>
                                    </XCUIElementTypeOther>
                                </XCUIElementTypeOther>
                            </XCUIElementTypeOther>
                        </XCUIElementTypeOther>
                    </XCUIElementTypeOther>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    </XCUIElementTypeWebView>
    .....
    willOpenChainList = [
        {
            "tag": "XCUIElementTypeWebView",
            "attrs": {"visible": "true", "enabled": "true", "type": "XCUIElementTypeWebView"}
        },
        {
            "tag": "XCUIElementTypeOther",
            "attrs": {"visible": "true", "enabled": "true", "type": "XCUIElementTypeOther"}
        },
        {
            "tag": "XCUIElementTypeStaticText",
            "attrs": {"visible": "true", "enabled": "true", "type": "XCUIElementTypeStaticText"}
        },
    ]
    willOpenSoup = CommonUtils.bsChainFind(soup, willOpenChainList)
    if willOpenSoup:
        parentOtherSoup = willOpenSoup.parent
        parentParentOtherSoup = parentOtherSoup.parent
        if parentParentOtherSoup:
```

```
continueOpenSoup = parentParentOtherSoup.find(  
    "XCUIElementTypeButton",  
    attrs={"visible":"true", "enabled":"true",  
}  
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 使用我的地理位置：
    <XCUIElementTypeOther type="XCUIElementTypeOther">
        <XCUIElementTypeNavigationBar type="XCUIElementTypeNavigation">
            <XCUIElementTypeButton type="XCUIElementTypeTextElement">
                <XCUIElementTypeStaticText type="XCUIElementTypeTextElement">
                    </XCUIElementTypeNavigationBar>
                <XCUIElementTypeOther type="XCUIElementTypeTable">
                    <XCUIElementTypeOther type="XCUIElementTypeCell">
                        <XCUIElementTypeOther type="XCUIElementTypeTable">
                            <XCUIElementTypeStaticText type="XCUIElementTypeTextElement">
                                <XCUIElementTypeCell type="XCUIElementTypeCell">
                                    <XCUIElementTypeOther type="XCUIElementTypeTextElement">
                                        <XCUIElementTypeStaticText type="XCUIElementTypeTextElement">
                                            <XCUIElementTypeCell type="XCUIElementTypeCell">
                                                <XCUIElementTypeSwitch type="XCUIElementTypeTextElement">
                                                    </XCUIElementTypeCell>
                                                <XCUIElementTypeOther type="XCUIElementTypeTextElement">
                                                    </XCUIElementTypeTable>
                                                </XCUIElementTypeCell>
                                            </XCUIElementTypeTable>
                                        </XCUIElementTypeCell>
                                    </XCUIElementTypeTextElement>
                                </XCUIElementTypeCell>
                            </XCUIElementTypeTextElement>
                        </XCUIElementTypeCell>
                    </XCUIElementTypeTable>
                </XCUIElementTypeCell>
            </XCUIElementTypeTextElement>
        </XCUIElementTypeNavigation">
    </XCUIElementTypeOther>
    .....
    foundAndProcessedPopup = False
    useLocationSwitchQuery = {"type": "XCUIElementTypeSwitch"}
    parentCellClassChain = "/XCUIElementTypeCell[`enabled`"
    useLocationSwitchQuery["parent_class_chains"] = [ parentCellClassChain ]
    foundAndClickeUseLocation = self.findAndClickElement(useLocationSwitchQuery)
    logging.debug("foundAndClickeUseLocation=%s", foundAndClickeUseLocation)
    if foundAndClickeUseLocation:
        parentNavBarClassChain = '/XCUIElementTypeNavigationBar[`enabled`'
        backButtonQuery = {"type": "XCUIElementTypeButton"}  
.....
        backButtonQuery["parent_class_chains"] = parentNavBarClassChain
        foundAndClickeBack = self.findAndClickElement(backButtonQuery)
        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">
        <XCUIElementTypeOther type="XCUIElementTypeTable">
            <XCUIElementTypeButton type="XCUIElementTypeTextFiled">
                <XCUIElementTypeStaticText type="XCUIElementTypeText">
                    <XCUIElementTypeButton type="XCUIElementTypeText">
                </XCUIElementTypeText>
            </XCUIElementTypeTable>
            <XCUIElementTypeOther type="XCUIElementTypeCell">
                <XCUIElementTypeCell type="XCUIElementTypeText">
                    <XCUIElementTypeStaticText type="XCUIElementTypeText">
                        <XCUIElementTypeImage type="XCUIElementTypeText">
                    </XCUIElementTypeText>
                </XCUIElementTypeCell>
            </XCUIElementTypeOther>
        </XCUIElementTypeTable>
    </XCUIElementTypeOther>
    .....
    networkNotStableChainList = [
        {
            "tag": "XCUIElementTypeTable",
            "attrs": {"enabled": "true", "visible": "true", "text": "京东金融 网络不稳定 请点击重试:"}
        },
        {
            "tag": "XCUIElementTypeCell",
            "attrs": {"enabled": "true", "visible": "true", "text": "京东金融 网络不稳定 请点击重试:"}
        },
        {
            "tag": "XCUIElementTypeStaticText",
            "attrs": {"enabled": "true", "visible": "true", "text": "京东金融 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">
            <XCUIElementTypeTable type="XCUIElementTypeTable">
                <XCUIElementTypeOther type="XCUIElementTypeOther">
                    <XCUIElementTypeImage type="XCUIElementTypeImage">
                    <XCUIElementTypeStaticText type="XCUIElementTypeText">
                    <XCUIElementTypeButton type="XCUIElementTypeButton">
                    </XCUIElementTypeImage>
                </XCUIElementTypeOther>
            </XCUIElementTypeTable>
        </XCUIElementTypeOther>
    .....
    tryRefreshChainList = [
        {
            "tag": "XCUIElementTypeOther",
            "attrs": {"enabled": "true", "visible": "true", "type": "XCUIElementTypeTable"}
        },
        {
            "tag": "XCUIElementTypeOther",
            "attrs": {"enabled": "true", "visible": "true", "type": "XCUIElementTypeImage"}
        },
        {
            "tag": "XCUIElementTypeButton",
            "attrs": {"enabled": "true", "visible": "true", "type": "XCUIElementTypeText"}
        },
    ]
    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", "label": "刷新"}
    # 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">
        <XCUIElementTypeStaticText type="XCUIElementTypeText">
            </XCUIElementTypeStaticText>
    </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}
    )
    if foundTouchReload:
        self.clickElementCenterPosition(foundTouchReload)
        foundAndProcessed = True

    if not foundAndProcessed:
        .....
        <XCUIElementTypeButton type="XCUIElementTypeButton">
            <XCUIElementTypeStaticText type="XCUIElementTypeText">
                </XCUIElementTypeStaticText>
        </XCUIElementTypeButton>

        <XCUIElementTypeWindow type="XCUIElementTypeWindow">
            <XCUIElementTypeOther type="XCUIElementTypeWindow">
                <XCUIElementTypeOther type="XCUIElementTypeWindow">
                    <XCUIElementTypeOther type="XCUIElementTypeWindow">
                        <XCUIElementTypeButton type="XCUIElementTypeButton">
                            <XCUIElementTypeStaticText type="XCUIElementTypeText">
                                </XCUIElementTypeStaticText>
                            <XCUIElementTypeStaticText type="XCUIElementTypeText">
                                </XCUIElementTypeStaticText>
                            <XCUIElementTypeStaticText type="XCUIElementTypeText">
                                </XCUIElementTypeStaticText>
                            <XCUIElementTypeStaticText type="XCUIElementTypeText">
                                </XCUIElementTypeStaticText>
                        </XCUIElementTypeButton>
                        </XCUIElementTypeOther>
                    </XCUIElementTypeOther>
                    </XCUIElementTypeOther>
                </XCUIElementTypeWindow>
            </XCUIElementTypeOther>
            .....

            securityWarningChainList = self.generateCommonPopups()
            securityWarningSoup = CommonUtils.bsChainFind(soup,
            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"}
                    )
                    if foundCertProblem:
                        self.clickElementCenterPosition(foundCertProblem)
                        foundAndProcessed = True

```

```
        foundCancel = parentButtonSoup.find(  
            "XCUIElementTypeStaticText",  
            attrs={"visible":"true", "enabled":  
        })  
        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"
        <XCUIElementTypeOther type="XCUIElementTypeOther"
            <XCUIElementTypeOther type="XCUIElementTypeOther"
                <XCUIElementTypeOther type="XCUIElementTypeTable"
                    <XCUIElementTypeOther type="XCUIElementTypeCell"
                        <XCUIElementTypeStaticText type="XCUIElementTypeText">
                            <XCUIElementTypeOther type="XCUIElementTypeImage"/>
                            <XCUIElementTypeStaticText type="XCUIElementTypeText">
                                <XCUIElementTypeOther type="XCUIElementTypeTable">
                                    <XCUIElementTypeCell type="XCUIElementTypeCell">
                                        <XCUIElementTypeImage type="XCUIElementTypeImage"/>
                                        <XCUIElementTypeStaticText type="XCUIElementTypeText">
                                            <XCUIElementTypeText>
                                                </XCUIElementTypeText>
                                            </XCUIElementTypeStaticText>
                                        </XCUIElementTypeCell>
                                    </XCUIElementTypeTable>
                                </XCUIElementTypeOther>
                            </XCUIElementTypeStaticText>
                        </XCUIElementTypeCell>
                    </XCUIElementTypeTable>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
    </XCUIElementTypeWindow>
    .....
    weixinLoginChainList = self.generateCommonPopupItemChain(
        secondLevelTag="XCUIElementTypeOther",
        thirdLevelValue="微信登录"
    )
    foundWeixinLogin = CommonUtils.bsChainFind(soup, weixinLoginChainList)

    tableChainList = self.generateCommonPopupItemChain(
        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",
    })
    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"
        <XCUIElementTypeOther type="XCUIElementTypeOther"
            <XCUIElementTypeOther type="XCUIElementTypeOther"
                <XCUIElementTypeOther type="XCUIElementTypeOther"
                    <XCUIElementTypeButton type="XCUIElementTypeButton"
                        <XCUIElementTypeStaticText type="XCUIElementTypeStaticText"
                            <XCUIElementTypeStaticText type="XCUIElementTypeStaticText"
                                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText"
                                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText"
                                        </XCUIElementTypeButton>
                                    </XCUIElementTypeOther>
                                </XCUIElementTypeOther>
                            </XCUIElementTypeOther>
                        </XCUIElementTypeButton>
                    </XCUIElementTypeOther>
                </XCUIElementTypeOther>
            </XCUIElementTypeOther>
        </XCUIElementTypeWindow>
    .....
    leaveWeixinChainList = self.generateCommonPopupItemChain()
    foundLeaveWeixin = CommonUtils.bsChainFind(soup, leaveWeixinChainList)
    if foundLeaveWeixin:
        parentButtonSoup = foundLeaveWeixin.parent
        if parentButtonSoup:
            foundCancel = parentButtonSoup.find(
                "XCUIElementTypeStaticText",
                attrs={"visible": "true", "enabled": "true",
            )
            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="XCUIElementTypeWi
        . . .
        <XCUIElementTypeButton type="XCUIElementTypeBu
            <XCUIElementTypeStaticText type="XCUIE
            <XCUIElementTypeStaticText type="XCUIE
            <XCUIElementTypeStaticText type="XCUIE
        </XCUIElementTypeButton>
        <XCUIElementTypeOther type="XCUIElementTypeO
            <XCUIElementTypeOther type="XCUIElement
                <XCUIElementTypeButton type="XCUIE
                <XCUIElementTypeButton type="XCUIE
            </XCUIElementTypeOther>
        </XCUIElementTypeOther>
        . . .
    </XCUIElementTypeWindow>

    微信登录失败:
    <XCUIElementTypeWindow type="XCUIElementTypeWi
        <XCUIElementTypeOther type="XCUIElementTypeO
            <XCUIElementTypeButton type="XCUIElement
                <XCUIElementTypeOther type="XCUIEl
                    <XCUIElementTypeImage type="XCI
                    <XCUIElementTypeStaticText type=
                    <XCUIElementTypeStaticText type=
                    <XCUIElementTypeButton type="XC
                </XCUIElementTypeOther>
            </XCUIElementTypeButton>
        </XCUIElementTypeOther>
    </XCUIElementTypeWindow>

    小程序:
        获取信息失败:
        <XCUIElementTypeButton type="XCUIElementTypeBu
            <XCUIElementTypeOther type="XCUIElementTypeO
                <XCUIElementTypeImage type="XCUIEl
                <XCUIElementTypeStaticText type="XCUIE
                <XCUIElementTypeTextView type="XCUIEl
                <XCUIElementTypeButton type="XCUIEl
                <XCUIElementTypeButton type="XCUIEl
            </XCUIElementTypeOther>
        </XCUIElementTypeButton>
        .....
    # getInfoFailChainList = self.generateCommonPopupItemC
```

```
# getInfoFailSoup = CommonUtils.bsChainFind(soup, getInfoFail)
# if getInfoFailSoup:
#     parentButtonOrOtherSoup = getInfoFailSoup.parent
doSomeFailP = re.compile("\$+失败\$") # 获取信息失败, 微信弹框
doSomeFailChainList = self.generateCommonPopupItemChain()
doSomeFailSoup = CommonUtils.bsChainFind(soup, doSomeFailP)
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",
        )

        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",
            )

        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">
            <XCUIElementTypeOther type="XCUIElementTypeOther">
                <XCUIElementTypeImage type="XCUIElementTypeImage">
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText">
                <XCUIElementTypeTextView type="XCUIElementTypeTextView">
                <XCUIElementTypeButton type="XCUIElementTypeButton">
                <XCUIElementTypeButton type="XCUIElementTypeButton">
            </XCUIElementTypeOther>
        </XCUIElementTypeButton>
    .....
    authorizeLocationChainList = [
        {
            "tag": "XCUIElementTypeButton",
            "attrs": {"visible": "true", "enabled": "true", "name": "同意"}
        },
        {
            "tag": "XCUIElementTypeOther",
            "attrs": {"visible": "true", "enabled": "true", "name": "同意"}
        },
        {
            "tag": "XCUIElementTypeStaticText",
            "attrs": {"visible": "true", "enabled": "true", "name": "同意"}
        },
    ]
    authorizeLocationSoup = CommonUtils.bsChainFind(soup, authorizeLocationChainList)
    if authorizeLocationSoup:
        parentOtherSoup = authorizeLocationSoup.parent
        confirmSoup = parentOtherSoup.find(
            "XCUIElementTypeButton",
            attrs={"visible": "true", "enabled": "true", "name": "同意"}
        )
        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
    """
    foundAndProcessedPopup = False
    """
        微信-小程序 弹框 警告 尚未进行授权:
        <XCUIElementTypeButton type="XCUIElementTypeButton"
            <XCUIElementTypeOther type="XCUIElementTypeOther"
                <XCUIElementTypeImage type="XCUIElementTypeImage"/>
                <XCUIElementTypeStaticText type="XCUIElementTypeStaticText"/>
                <XCUIElementTypeTextView type="XCUIElementTypeTextView"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton"/>
                <XCUIElementTypeButton type="XCUIElementTypeButton"/>
            </XCUIElementTypeOther>
        </XCUIElementTypeButton>
    """
    warningChainList = [
        {
            "tag": "XCUIElementTypeButton",
            "attrs": {"visible": "true", "enabled": "true", "name": "确定"}
        },
        {
            "tag": "XCUIElementTypeOther",
            "attrs": {"visible": "true", "enabled": "true", "name": "确定"}
        },
        {
            "tag": "XCUIElementTypeStaticText",
            "attrs": {"visible": "true", "enabled": "true", "name": "提示文字"}
        }
    ]
    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 / 小程序 弹框 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">
        <XCUIElementTypeOther type="XCUIElementTypeImage">
        <XCUIElementTypeOther type="XCUIElementTypeText">
        <XCUIElementTypeImage type="XCUIElementTypeImage">
        <XCUIElementTypeStaticText type="XCUIElementTypeText">
        <XCUIElementTypeStaticText type="XCUIElementTypeText">
        <XCUIElementTypeStaticText type="XCUIElementTypeText">
        <XCUIElementTypeTable type="XCUIElementTypeTable">
            <XCUIElementTypeCell type="XCUIElementTypeCell">
                <XCUIElementTypeOther type="XCUIElementTypeImage">
                <XCUIElementTypeOther type="XCUIElementTypeImage">
                <XCUIElementTypeImage type="XCUIElementTypeImage">
                <XCUIElementTypeOther type="XCUIElementTypeText">
                <XCUIElementTypeStaticText type="XCUIElementTypeText">
                <XCUIElementTypeStaticText type="XCUIElementTypeText">
                <XCUIElementTypeImage type="XCUIElementTypeImage">
            </XCUIElementTypeCell>
        </XCUIElementTypeTable>
        <XCUIElementTypeButton type="XCUIElementTypeText">
        <XCUIElementTypeButton type="XCUIElementTypeText">
        <XCUIElementTypeButton type="XCUIElementTypeText">
    </XCUIElementTypeOther>
</XCUIElementTypeOther>

    弹框 获取你的通讯地址：
    <XCUIElementTypeOther type="XCUIElementTypeOther">
        <XCUIElementTypeOther type="XCUIElementTypeImage">
        <XCUIElementTypeOther type="XCUIElementTypeText">
        <XCUIElementTypeImage type="XCUIElementTypeImage">
        <XCUIElementTypeStaticText type="XCUIElementTypeText">
        <XCUIElementTypeStaticText type="XCUIElementTypeText">
        <XCUIElementTypeStaticText type="XCUIElementTypeText">
        <XCUIElementTypeButton type="XCUIElementTypeText">
        <XCUIElementTypeButton type="XCUIElementTypeText">
    </XCUIElementTypeOther>
</XCUIElementTypeOther>

    小程序 弹框 需要获取你的地理位置：
    <XCUIElementTypeOther type="XCUIElementTypeOther">
        <XCUIElementTypeOther type="XCUIElementTypeImage">
        <XCUIElementTypeOther type="XCUIElementTypeText">
        <XCUIElementTypeImage type="XCUIElementTypeImage">
        <XCUIElementTypeOther type="XCUIElementTypeText">
        <XCUIElementTypeText type="XCUIElementTypeText">
        <XCUIElementTypeAlert type="XCUIElementTypeText">
```

```
<XCUIElementTypeOther type="XCUIElementTypeAlert">
    <XCUIElementTypeOther type="XCUIElementTypeOther">
        <XCUIElementTypeOther type="XCUIElementTypeOther">
            <XCUIElementTypeOther type="XCUIElementTypeOther">
                <XCUIElementTypeOther type="XCUIElementTypeOther">
                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                                    <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                                        <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                                            <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
                                                                                                                                <XCUIElementTypeOther type="XCUIElementTypeOther">
................................................................
# getYourP = re.compile("获取你的\S+") # 获取你的通讯地址,
getYourP = re.compile("(需要)?获取你的\S+)") # 获取你的通讯地址
getYourChainList = [
    {
        "tag": "XCUIElementTypeOther",
        "attrs": {"visible": "true", "enabled": "true", "label": "获取你的通讯地址", "type": "XCUIElementTypeOther", "text": "(需要)?获取你的通讯地址", "value": "(需要)?获取你的通讯地址"}
    },
    {
        "tag": "XCUIElementTypeOther",
        "attrs": {"visible": "true", "enabled": "true", "label": "输入你的通讯地址", "type": "XCUIElementTypeOther", "text": "输入你的通讯地址", "value": "输入你的通讯地址"}
    }
]
```

```
        "attrs": {"visible": "true", "enabled": "true", },
    },
    {
        "tag": "XCUIElementTypeStaticText",
        "attrs": {"visible": "true", "enabled": "true", },
    },
]
getYourSoup = CommonUtils.bsChainFind(soup, getByCharName)
if getYourSoup:
    parentOther = getYourSoup.parent
    parentParentOther = None
    parentParentParentOther = None

    allowTag = "XCUIElementTypeButton"
    allowAttrs = {"visible": "true", "enabled": "true", }

    # 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"
        <XCUIElementTypeOther type="XCUIElementTypeOther"
            <XCUIElementTypeButton type="XCUIElementTypeButton"
                <XCUIElementTypeOther type="XCUIElementTypeOther"
                    <XCUIElementTypeImage type="XCUIElementTypeImage"/>
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText"/>
                    <XCUIElementTypeStaticText type="XCUIElementTypeStaticText"/>
                    <XCUIElementTypeButton type="XCUIElementTypeButton"/>
                </XCUIElementTypeOther>
            </XCUIElementTypeButton>
        </XCUIElementTypeOther>
    </XCUIElementTypeWindow>
    .....
    disclaimerChainList = [
        {
            "tag": "XCUIElementTypeWindow",
            "attrs": {"visible": "true", "enabled": "true", "type": "XCUIElementTypeWindow"}
        },
        {
            "tag": "XCUIElementTypeButton",
            "attrs": {"visible": "true", "enabled": "true", "type": "XCUIElementTypeButton"}
        },
        {
            "tag": "XCUIElementTypeStaticText",
            "attrs": {"visible": "true", "enabled": "true", "type": "XCUIElementTypeStaticText"}
        }
    ]
    disclaimerSoup = CommonUtils.bsChainFind(soup, disclaimerChainList)
    if disclaimerSoup:
        parentOtherSoup = disclaimerSoup.parent
        if parentOtherSoup:
            foundIKnow = parentOtherSoup.find(
                "XCUIElementTypeButton",
                attrs={"visible": "true", "enabled": "true", "type": "XCUIElementTypeButton"})
            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": "full", "height": "full"}
    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: 2020-06-20 22:50:21

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
    """

    """
        途虎养车 左滑引导页 1页/3页：
        <XCUIElementTypeOther type="XCUIElementTypeOther">
            <XCUIElementTypeOther type="XCUIElementTypeOther">
                <XCUIElementTypeScrollView type="XCUIElementTypeScrollView">
                    <XCUIElementTypeImage type="XCUIElementTypeImage">
                </XCUIElementTypeScrollView>
            <XCUIElementTypeButton type="XCUIElementTypeButton">
                <XCUIElementTypeButton type="XCUIElementTypeButton">
                    <XCUIElementTypeButton type="XCUIElementTypeButton">
                </XCUIElementTypeButton>
            </XCUIElementTypeButton>
        </XCUIElementTypeOther>
    </XCUIElementTypeOther>

    途虎养车 左滑引导页 2页/3页：
    <XCUIElementTypeOther type="XCUIElementTypeOther">
        <XCUIElementTypeOther type="XCUIElementTypeOther">
            <XCUIElementTypeScrollView type="XCUIElementTypeScrollView">
                <XCUIElementTypeImage type="XCUIElementTypeImage">
            </XCUIElementTypeScrollView>
        <XCUIElementTypeButton type="XCUIElementTypeButton">
            <XCUIElementTypeButton type="XCUIElementTypeButton">
                <XCUIElementTypeButton type="XCUIElementTypeButton">
            </XCUIElementTypeButton>
        </XCUIElementTypeButton>
    </XCUIElementTypeOther>

    途虎养车 左滑引导页 3页/3页 立即进入：
    <XCUIElementTypeOther type="XCUIElementTypeOther">
        <XCUIElementTypeOther type="XCUIElementTypeOther">
            <XCUIElementTypeScrollView type="XCUIElementTypeScrollView">
                <XCUIElementTypeImage type="XCUIElementTypeImage">
            </XCUIElementTypeScrollView>
        <XCUIElementTypeButton type="XCUIElementTypeButton">
            <XCUIElementTypeButton type="XCUIElementTypeButton">
                <XCUIElementTypeButton type="XCUIElementTypeButton">
            </XCUIElementTypeButton>
        </XCUIElementTypeButton>
    </XCUIElementTypeOther>
    """

    # GuideText = "guide"
    # parentScrollViewClassChain = "/XCUIElementTypeScrollView"
    # guideImgeQuery = {"type": "XCUIElementTypeImage", "name": "guideText"}
    # guideImgeQuery["parent_class_chains"] = [parentScrollViewClassChain]
    # isFound, guideImgeElement = findElement(curSession, guideImgeQuery)
    # if isFound:
    #     # foundAndClicked = clickElement(curSession, guideImgeElement)
    #     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, guideP)
    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 #
            soupType = soupAttrDict.get("type") # 'XCUIElementTypeImage'
            soupName = soupAttrDict.get("name") # 'loading'
            soupVisible = soupAttrDict.get("visible") #
            isButton = soupType == "XCUIElementTypeButton"
            # isLoadingName = bool(re.search(soupName,
            # 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:

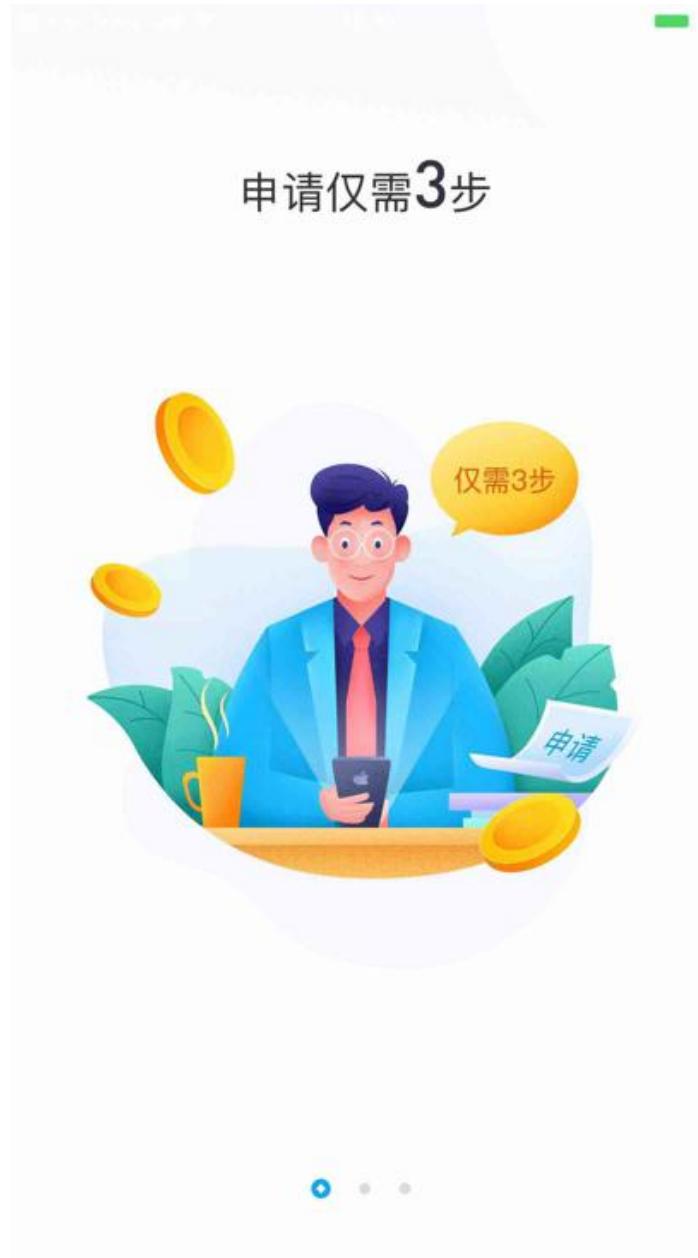
```

```
# end page, click button, into main page
lastPageButtonSoup = validButtonSoupList[0]
self.clickElementCenterPosition(lastPageButton)
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"

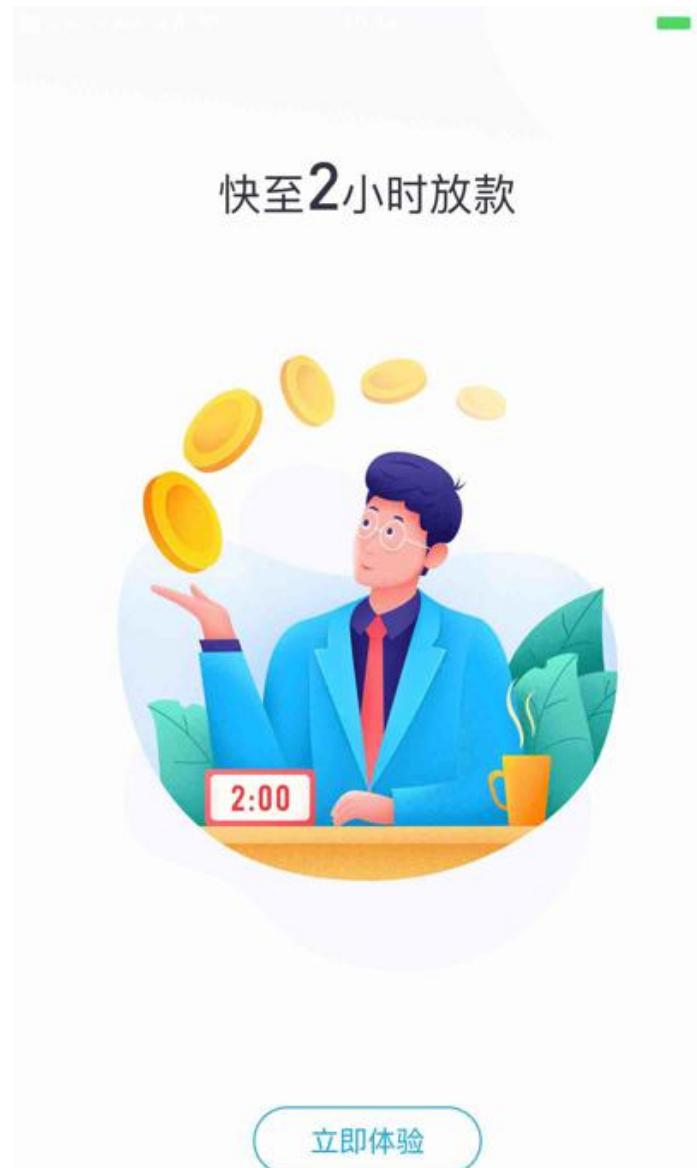
return foundAndSwipeGuideToMain
```

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: 2020-06-20 22:38:16

源码分析

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: 2020-06-20 22:00:14

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: 2020-06-01 18:17:10

文档

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: 2020-06-20 22:38:34

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](#)
- [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](#)
- [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: 2020-06-20 23:10:39](#)