

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
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: v1.0
- 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 22:01:27

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](#)

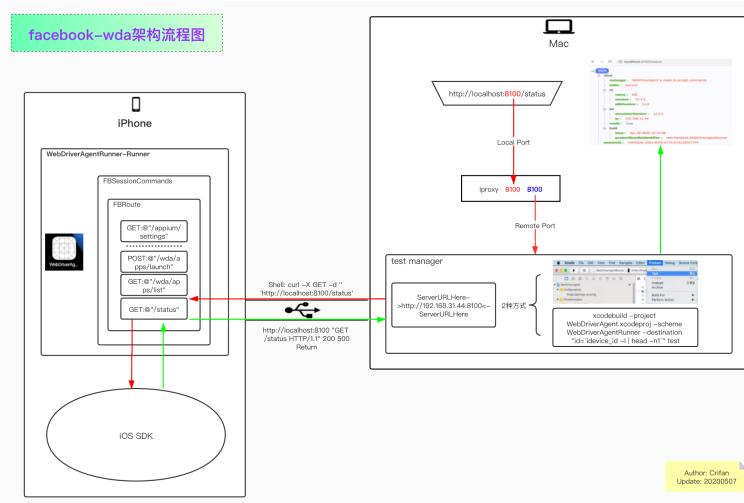
crifan.com, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved,
powered by Gitbook最后更新：2020-06-01 18:33:14

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" name="Text" value="Text" />
            <XCUIElementTypeButton type="XCUIElementTypeButton" name="Button" 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="XCUIElementTypeText" value="【善友筹】（已审核）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！" />
    <XCUIElementTypeProgressIndicator type="XCUIElementTypeProgressIndicator" value="14%" />
    <XCUIElementTypeStaticText type="XCUIElementTypeText" value="发起人:刘杰 认筹金额:2757 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！

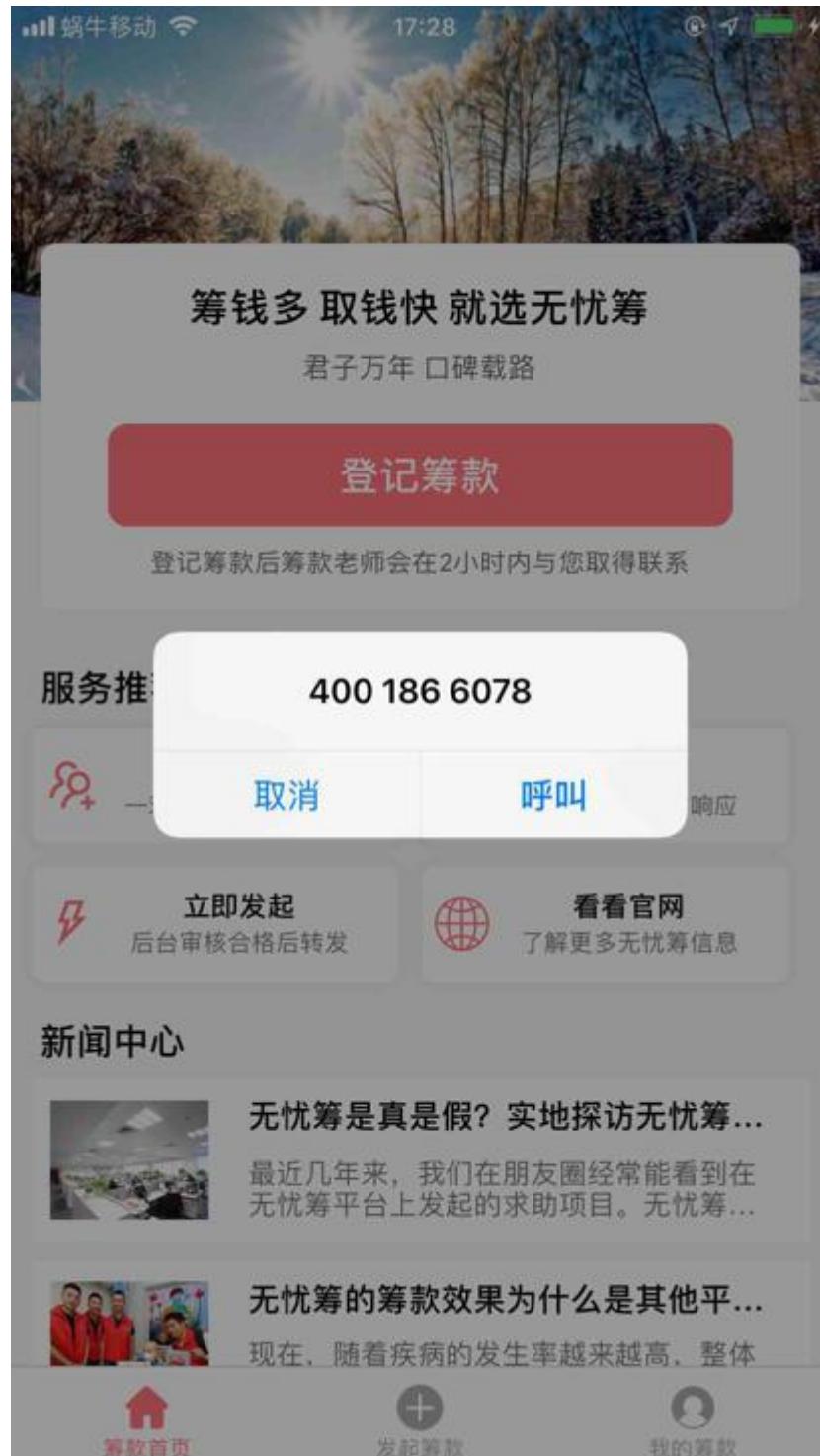
你妹的，那还怎么解析出有效节点，根本没法提取有效节点，和后续抓取。

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：



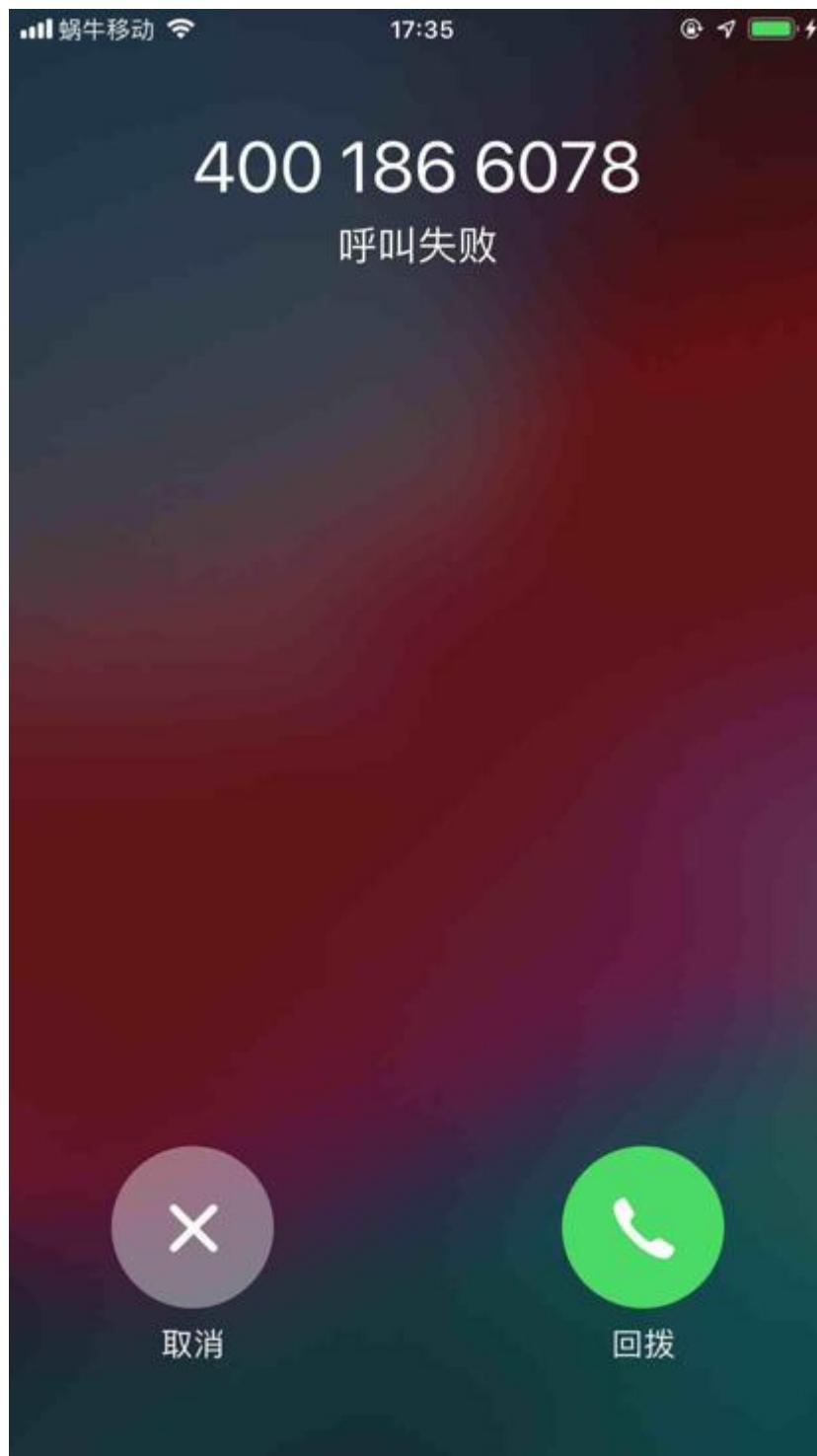
代码已找到了 取消 按钮，然后去点击 其中间坐标位置

```
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。

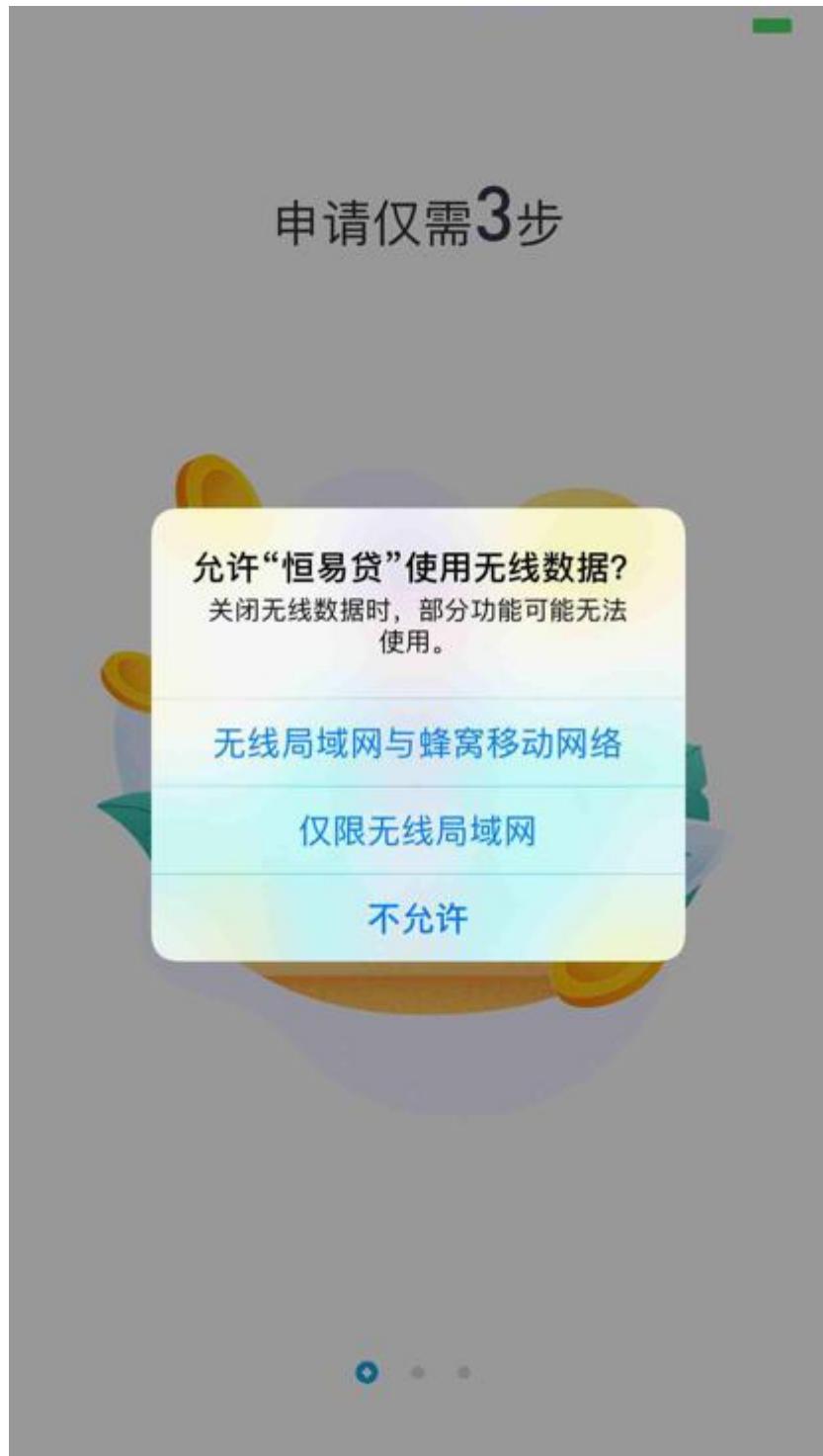
【后记2】

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：

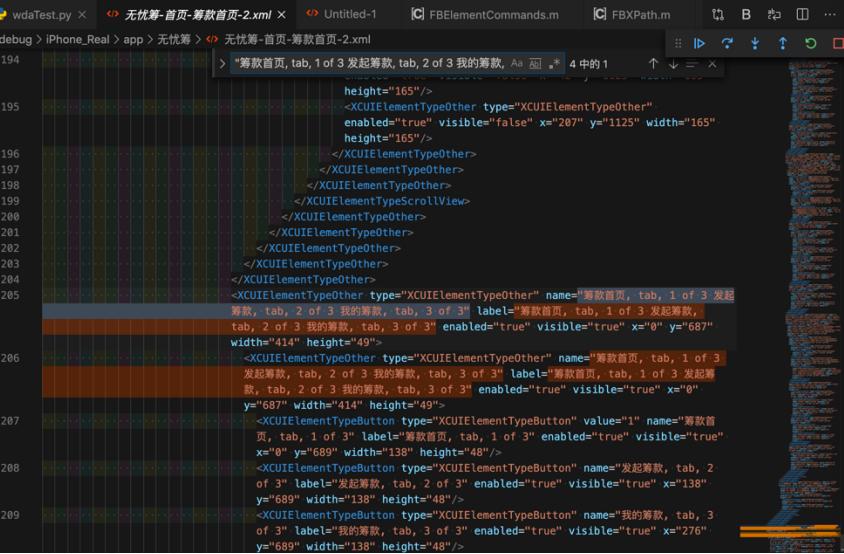


左下角的 3个tab页的父级元素，对应 locator，调试出现警告：

[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 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">
205     height="165"/>
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

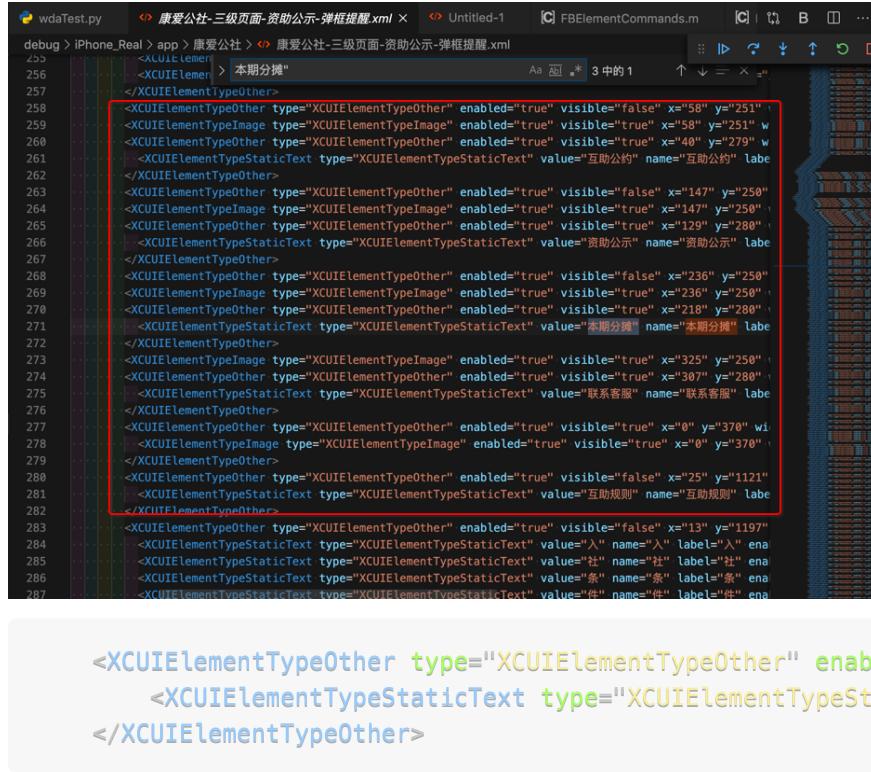


其中，正常的符合预期的是，页面xml源码中，有页面中的元素，比如顶部的第二排的 互助公约 资助公示 本期分摊 联系客服 等但是点击了 资助公示 后，进入 三级页面：
康爱公社-三级页面-资助公示-弹框提醒.jpg



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
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     <XCUIElement>
256         <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="58" y="251">
257             <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="58" y="251" w="140" h="140" label=""></XCUIElementTypeImage>
258             <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="147" y="250">
259                 <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="147" y="250" w="140" h="140" label=""></XCUIElementTypeImage>
260             <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="40" y="279" w="140" h="140" label=""></XCUIElementTypeImage>
261             <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="互助公约" name="互助公约" label=""></XCUIElementTypeText>
262         </XCUIElementTypeOther>
263         <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="236" y="250">
264             <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="236" y="250" w="140" h="140" label=""></XCUIElementTypeImage>
265             <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="147" y="280" w="140" h="140" label=""></XCUIElementTypeImage>
266             <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="资助公示" name="资助公示" label=""></XCUIElementTypeText>
267         </XCUIElementTypeOther>
268         <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="236" y="250">
269             <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="236" y="250" w="140" h="140" label=""></XCUIElementTypeImage>
270             <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="218" y="280" w="140" h="140" label=""></XCUIElementTypeImage>
271             <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="本期分摊" name="本期分摊" label=""></XCUIElementTypeText>
272         </XCUIElementTypeOther>
273         <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="325" y="250" w="140" h="140" label=""></XCUIElementTypeImage>
274         <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="307" y="280" w="140" h="140" label=""></XCUIElementTypeImage>
275         <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="联系客服" name="联系客服" label=""></XCUIElementTypeText>
276     </XCUIElementTypeOther>
277     <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="370" w="140" h="140" label=""></XCUIElementTypeImage>
278     <XCUIElementTypeImage type="XCUIElementTypeImage" enabled="true" visible="true" x="0" y="370" w="140" h="140" label=""></XCUIElementTypeImage>
279     <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="25" y="1121" w="140" h="140" label=""></XCUIElementTypeImage>
280     <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="互助规则" name="互助规则" label=""></XCUIElementTypeText>
281 </XCUIElementTypeImage>
282     <XCUIElementTypeOther type="XCUIElementTypeOther" enabled="true" visible="false" x="13" y="1197" w="140" h="140" label=""></XCUIElementTypeImage>
283     <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="入" name="入" label="入" enabled="true" visible="true" x="13" y="1197" w="140" h="140" label=""></XCUIElementTypeText>
284     <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="社" name="社" label="社" enabled="true" visible="true" x="13" y="1197" w="140" h="140" label=""></XCUIElementTypeText>
285     <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="条" name="条" label="条" enabled="true" visible="true" x="13" y="1197" w="140" h="140" label=""></XCUIElementTypeText>
286     <XCUIElementTypeStaticText type="XCUIElementTypeStaticText" value="件" name="件" label="件" enabled="true" visible="true" x="13" y="1197" w="140" h="140" label=""></XCUIElementTypeText>
287 </XCUIElementTypeImage>
```

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/康爱公社-三级页面-资助公示-弹框提醒.xml" enabled="true" visible="true" x="0" y="0" w="1000" h="1000" label=""></XCUIElementTypeOther>
```

常见问题和心得

```
wdaTest.py 康泰公社_三级页面-资助公告-弹框提醒.xml Untitled-1 FBElementCommands.m FBXPath.m _init_.py B
debug > iPhone_Real > app > 康泰公社 > 康泰公社_三级页面-资助公告-弹框提醒.xml
228 <XCUIElement type="XCUIElementTypeButton" ty="pages/" Aa * 6 中的 3 ↑ ↓ = ×
229 <XCUIElement type="XCUIElementTypeButton" name="康泰" label="康泰" enabled="true" visible="false" type="XCU
230 <XCUIElement type="XCUIElementTypeButton" name="互助社" label="互助社" enabled="true" visible="false" type="XCU
231 <XCUIElement type="XCUIElementTypeButton" name="透明" label="透明" enabled="true" visible="false" type="XCU
232 </XCUIElement type="XCUIElementTypeOther">
233 </XCUIElement type="XCUIElementTypeOther">
234 </XCUIElement type="XCUIElementTypeOther">
235 <XCUIElement type="XCUIElementTypeOther" type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="0" width="414" height="73
236 <XCUIElement type="XCUIElementTypeOther" type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="0" width="414" height="141
237 <XCUIElement type="XCUIElementTypeOther" type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="0" width="414" height="141
238 <XCUIElement type="XCUIElementTypeButton" type="XCUIElementTypeButton" enabled="true" visible="false" x="0" y="20" width="411" heig
239 <XCUIElement type="XCUIElementTypeOther" type="XCUIElementTypeOther" enabled="true" visible="false" x="46" y="20" width="322" heig
240 </XCUIElement type="XCUIElementTypeOther">
241 <XCUIElement type="XCUIElementTypeOther" type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="64" width="414" height="141
242 <XCUIElement type="XCUIElementTypeWebView" type="XCUIElementTypeWebView" enabled="true" visible="false" x="0" y="64" width="414" height="141
243 <XCUIElement type="XCUIElementTypeOther" type="XCUIElementTypeOther" enabled="true" visible="false" x="0" y="64" width="414" height="141
244 <XCUIElement type="XCUIElementTypeOther" type="XCUIElementTypeOther" type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="141
245 <XCUIElement type="XCUIElementTypeOther" type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="141
246 <XCUIElement type="XCUIElementTypeOther" type="XCUIElementTypeOther" type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="141
247 <XCUIElement type="XCUIElementTypeOther" type="XCUIElementTypeOther" type="XCUIElementTypeOther" type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="141
248 <XCUIElement type="XCUIElementTypeOther" type="XCUIElementTypeOther" type="XCUIElementTypeOther" type="XCUIElementTypeOther" type="XCUIElementTypeOther" enabled="true" visible="true" x="0" y="0" width="414" height="141
249 <XCUIElement type="XCUIElementTypeImage" type="XCUIElementTypeImage" enabled="true" visible="true" x="13" y="109" width="10" height="10
250 </XCUIElement type="XCUIElementTypeOther">
251 <XCUIElement type="XCUIElementTypeOther" type="XCUIElementTypeText" type="XCUIElementTypeStaticText" value="百万医保补充互助社[报销型]" name="百万医保补充互助社[报销型]"
252 </XCUIElement type="XCUIElementTypeOther">
253 <XCUIElement type="XCUIElementTypeOther" type="XCUIElementTypeText" type="XCUIElementTypeStaticText" value="加入" name="加入" label="加入"
254 <XCUIElement type="XCUIElementTypeText" type="XCUIElementTypeStaticText" value="加入" name="加入" label="加入"
255 <XCUIElement type="XCUIElementTypeText" type="XCUIElementTypeStaticText" value="加入" name="加入" label="加入"
256 <XCUIElement type="XCUIElementTypeText" type="XCUIElementTypeStaticText" value="加入" name="加入" label="加入"
257 <XCUIElement type="XCUIElementTypeText" type="XCUIElementTypeStaticText" value="加入" name="加入" label="加入"
258 <XCUIElement type="XCUIElementTypeText" type="XCUIElementTypeStaticText" value="加入" name="加入" label="加入"
259 <XCUIElement type="XCUIElementTypeText" type="XCUIElementTypeStaticText" value="加入" name="加入" label="加入"
```

且不止一个：



康健公社-三级页面-资助公示-弹框提醒.xml — iOSAutomation

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

pages/

```
As Aa * 6 中的 5
```

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

```
4/6 </XUIElementTypePageView>
```

```
471 </XUIElementTypePageOther>
```

```
472 </XUIElementTypePageOther>
```

```
473 <-XUIElementTypePageOther type="XUIElementTypePageOther" enabled="true" visible="false" x="0" y="0" width="414" height="736">
```

```
474 <-XUIElementTypePageOther type="XUIElementTypePageOther" enabled="true" visible="false" x="0" y="0" width="414" height="64">
```

```
475 <-XUIElementTypePageOther type="XUIElementTypePageOther" enabled="true" visible="false" x="0" y="0" width="414" height="20">
```

```
476 <-XUIElementTypePageButton type="XUIElementTypePageButton" enabled="true" visible="false" x="0" y="20" width="414" height="44">
```

```
477 <-XUIElementTypePageOther type="XUIElementTypePageOther" enabled="true" visible="false" x="46" y="20" width="322" height="44">
```

```
478 </XUIElementTypePageOther>
```

```
479 <-XUIElementTypePageOther type="XUIElementTypePageOther" enabled="true" visible="false" x="0" y="64" width="414" height="672">
```

```
480 <-XUIElementTypePageOther type="XUIElementTypePageOther" enabled="true" visible="false" x="0" y="64" width="414" height="50">
```

```
481 <-XUIElementTypePageView type="XUIElementTypePageView" enabled="true" visible="false" x="0" y="64" width="414" height="144">
```

```
482 <-XUIElementTypePageOther type="XUIElementTypePageOther" enabled="true" visible="false" x="0" y="64" width="414" height="67">
```

```
483 <-XUIElementTypePageOther type="XUIElementTypePageOther" enabled="true" visible="false" x="0" y="0" width="414" height="73">
```

```
484 <-XUIElementTypePageOther type="XUIElementTypePageOther" enabled="true" visible="true" x="0" y="0" width="414" height="414">
```

```
485 <-XUIElementTypePageOther type="XUIElementTypePageOther" enabled="true" visible="false" x="0" y="64" width="414" height="414">
```

```
486 <-XUIElementTypePageOther type="XUIElementTypePageOther" name="pages/home/notic[5]" label="pages/home/notic[5]" e="
```

```
487 <-XUIElementTypePageOther type="XUIElementTypePageOther" enabled="true" visible="true" x="0" y="0" width="287" height="144">
```

```
488 <-XUIElementTypeStaticText type="XUIElementTypeStaticText" value="长期资助" name="长期资助" label="长期资助">
```

```
489 </XUIElementTypePageOther>
```

```
490 <-XUIElementTypePageOther type="XUIElementTypePageOther" enabled="true" visible="true" x="207" y="64" width="207">
```

```
491 <-XUIElementTypeStaticText type="XUIElementTypeStaticText" value="历史资助" name="历史资助" label="历史资助">
```

```
492 </XUIElementTypePageOther>
```

```
493 <-XUIElementTypePageOther type="XUIElementTypePageOther" enabled="true" visible="true" x="0" y="18" width="414" height="144">
```

```
494 <-XUIElementTypeStaticText type="XUIElementTypeStaticText" value="筹款日期 2020 年 05 月 15 日" name="筹款日">
```

```
495 </XUIElementTypePageOther>
```

```
496 <-XUIElementTypePageOther type="XUIElementTypePageOther" enabled="true" visible="true" x="15" y="174" width="691" height="144">
```

```
497 <-XUIElementTypeImage type="XUIElementTypeImage" enabled="true" visible="true" x="15" y="176" width="551" height="144">
```

```
498 </XUIElementTypePageOther>
```

```
499 <-XUIElementTypePageOther type="XUIElementTypePageOther" enabled="true" visible="true" x="84" y="173" width="302" height="144">
```

```
500 <-XUIElementTypeStaticText type="XUIElementTypeStaticText" value="摩新芳" name="摩新芳" label="摩新芳" ena="
```

```
501 </XUIElementTypePageOther>
```

```
502 <-XUIElementTypePageOther type="XUIElementTypePageOther" enabled="true" visible="true" x="84" y="198" width="302" height="144">
```

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": "close"}
    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',
    # 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

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 10:23:23

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

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](#)
-
- [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: 2020-06-02 15:08:55