在Android端，appium基于WebDriver，并利用Bootstrap.js，最后通过调⽤用UiAutomator的命

令，实现App的自动化测试。UiAutomator测试框架是Android SDK⾃自带的App UI自动化测

试Java库。

另外由于UiAutomator对H5的⽀支持有限，appium引入了chromedriver来实现基于H5的自动

化。

主要原理：

1、左边的WebDriver script是我们的selenium测试脚本

2、中间是起的Appium的服务，Appium在这边起了一个Server（4723端口），跟selenium

Webdriver测试框架类似，Appium⽀支持标准的WebDriver JSONWireProtocol 。在这里提供

了一套web服务，Appium Server接收web driver 标准请求，解析请求内容，调⽤用对应的框

架响应操作。

如：脚本发送一个点击按钮的请求给appium server，

3、 appium server会把请求转发给中间件Bootstrap.jar ，它是用java写的，安装在手机上.

Bootstrap 接收appium 的命令(4724端口)，最终通过调⽤用UiAutomator的命令来实现。

4、最后执⾏行的结果由Bootstrap返回给appium server

5、另外，appium还用到了chromedriver来⽀支持基于H5（webview）的测试。

appium局限性

如果你在windows上安装appium，你没法使用预编译用于OS X的.app文件，你也将不能测试iOS apps，因为appium依赖OS X专用的库来支持iOS测试。这意味着你只能通过在mac上来运行iOS的app测试。这点限制挺大。

也就是说，在Windows下我们只能测试android，没法测试iOS系统。

作者：by小杰

链接：https://www.jianshu.com/p/3751cb7f148b

来源：简书

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

Appium is an open source, cross-platform test automation tool for **native, hybrid and mobile web and desktop apps**. We support **simulators (iOS), emulators (Android)**, and real devices (**iOS, Android, Windows, Mac**).

Want to skip straight to the action? Check out our [getting started](https://github.com/appium/appium/blob/master/docs/en/about-appium/getting-started.md) doc.

**Supported Platforms**

Appium supports app automation across a variety of platforms, like iOS, Android, and Windows. Each platform is supported by one or more "drivers", which know how to automate that particular platform. Choose a driver below for specific information about how that driver works and how to set it up:

* iOS
  + The [XCUITest Driver](https://github.com/appium/appium/blob/master/docs/en/drivers/ios-xcuitest.md)
  + (DEPRECATED) The [UIAutomation Driver](https://github.com/appium/appium/blob/master/docs/en/drivers/ios-uiautomation.md)
* Android
  + The [Espresso Driver](https://github.com/appium/appium/blob/master/docs/en/drivers/android-espresso.md)
  + The [UiAutomator2 Driver](https://github.com/appium/appium/blob/master/docs/en/drivers/android-uiautomator2.md)
  + (DEPRECATED) The [UiAutomator Driver](https://github.com/appium/appium/blob/master/docs/en/drivers/android-uiautomator.md)
* The [Windows Driver](https://github.com/appium/appium/blob/master/docs/en/drivers/windows.md) (for Windows Desktop apps)
* The [Mac Driver](https://github.com/appium/appium/blob/master/docs/en/drivers/mac.md) (for Mac Desktop apps)

[XCUITest Driver](https://github.com/appium/appium/blob/master/docs/en/drivers/ios-xcuitest.md)环境配置：

* iOS
* {
* "automationName": "XCUITest",
* "platformName": "iOS",
* "platformVersion": "12.2",
* "deviceName": "iPhone 8",
* ...
* }

UiAutomator2 driver

Android versions >= 5.0 (Lollipop, API level 21).

设置 automationName = UiAutomator2

 platformName (=Android), platformVersion, deviceName, and app capabilities

It is highly recommended to also set the appPackage and appActivity capabilities

web tests时需配置：

 app=””

browserName = Chrome.

emulator/device安装有chrome, 且版本和chromedriver对应.，电脑装的版本>=手机安装的chrome版本

 UiAutomator driver环境配置:

Jdk1.7

Appium versions < 1.14.0 不用设置automationName

Appium versions >= 1.14.0 设置automationName= UiAutomator1

Web应用测试时需配置：

设置app =””

设置browserName = Chrome

Espresso Driver 环境配置：

Jdk1.7

automationName = Espresso

## Windows Driver环境配置

Win10以上

可以进入管理员模式

开发模式打开

管理员模式启动apps

platformName=Windows

deviceName =WindowsPC

app=appID 或path路径

#### Universal Windows Platform App Testing

To test a UWP app, you can use any Selenium supported language and simply specify the **Application Id** for the app under test in the **app** capabilities entry. Below is an example of creating a test session for Windows **Alarms & Clock** app written in C#:

// Launch the AlarmClock app

DesiredCapabilities appCapabilities = new DesiredCapabilities();

appCapabilities.SetCapability("app", "Microsoft.WindowsAlarms\_8wekyb3d8bbwe!App");

AlarmClockSession = new WindowsDriver<WindowsElement>(new Uri("http://127.0.0.1:4723"), appCapabilities);

// Control the AlarmClock app

AlarmClockSession.FindElementByAccessibilityId("AddAlarmButton").Click();

AlarmClockSession.FindElementByAccessibilityId("AlarmNameTextBox").Clear();

#### Classic Windows App Testing

To test a classic Windows app, you can also use any Selenium supported language and specify the **full executable path** for the app under test in the **app** capabilities entry. Below is an example of creating a test session for Windows **Notepad** app:

// Launch Notepad

DesiredCapabilities appCapabilities = new DesiredCapabilities();

appCapabilities.SetCapability("app", @"C:\Windows\System32\notepad.exe");

NotepadSession = new WindowsDriver<WindowsElement>(new Uri("http://127.0.0.1:4723"), appCapabilities);

// Control the AlarmClock app

NotepadSession.FindElementByClassName("Edit").SendKeys("This is some text");

| **Locator Strategy** | **Matched Attribute** |
| --- | --- |
| accessibility id | AutomationId |
| class name | ClassName |
| name | Name |

package Appium;

import org.testng.annotations.Test;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeTest;

import io.appium.java\_client.android.AndroidDriver;

import io.appium.java\_client.AppiumDriver;

import java.net.MalformedURLException;

import java.net.URL;

import java.util.Set;

import org.openqa.selenium.By;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.remote.DesiredCapabilities;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.Select;

import org.openqa.selenium.support.ui.WebDriverWait;

public class WebViewSelendroidApp {

public static AndroidDriver<WebElement> driver;

public static AppiumDriver<WebElement> \_driver;

public static DesiredCapabilities cap = new DesiredCapabilities();

@BeforeTest

public void startAppium() throws MalformedURLException, InterruptedException{

System.out.println("setUP() :: driver.AndroidDriver() executed");

cap.setCapability("platformVersion","4.4.4");

cap.setCapability("platformName","Android");

cap.setCapability("deviceName","ZX1B32HNBD");

cap.setCapability("app","C:\\Users\\WebView-test.apk");

driver = new AndroidDriver<WebElement>(new URL("https://127.0.0.1:4725/wd/hub"),cap);

}

@Test

public void AppLogin() throws InterruptedException{

System.out.println("AppLogin() :: driver.start() executed");

By webView = By.className("android.webkit.WebView");

By title = By.id("android:id/title");

WebDriverWait wait = new WebDriverWait(driver,300);

driver.findElement(title).getText();

Set<String> availableContexts1 = driver.getContextHandles();

System.out.println("Total No of Context Found Before reaching WebView = "+ availableContexts1.size());

System.out.println("Context Name is "+ availableContexts1);

//4.1 Navigate to a portion of your app where a web view is active

driver.findElement(By.id("com.mkyong.android:id/buttonUrl")).click();

wait.until(ExpectedConditions.visibilityOfElementLocated(webView));

// 4.2 Call getContext() method which will returns a list of contexts we can access, like 'NATIVE\_APP' or 'WEBVIEW\_1'

Set<String> availableContexts = driver.getContextHandles();

System.out.println("Total No of Context Found After we reach to WebView = "+ availableContexts.size());

for(String context : availableContexts) {

if(context.contains("WEBVIEW")){

System.out.println("Context Name is " + context);

// 4.3 Call context() method with the id of the context you want to access and change it to WEBVIEW\_1

//(This puts Appium session into a mode where all commands are interpreted as being intended for automating the web view)

driver.context(context);

break;

}

}

String input\_box\_text = driver.findElement(By.id("name\_input")).getAttribute("value");

System.out.println("Pre written text inside text box is " + input\_box\_text);

driver.findElement(By.id("name\_input")).clear();

driver.findElement(By.id("name\_input")).sendKeys("Amit Jain"); System.out.println("No of dropdown on page "+ driver.findElements(By.xpath("//select")).size());

int size=driver.findElements(By.xpath("//select")).get(0).findElements(By.xpath("//option")).size();

System.out.println("No of Elements in dropdown "+ size);

WebElement car = driver.findElement(By.name("car"));

Select preferedCar=new Select(car);

preferedCar.selectByIndex(2);

System.out.println("Button Value is : " + driver.findElement(By.xpath("/html/body/form/div/input[2]")).getAttribute("value"));

//Key code constant: Back key.

//Constant Value: 4 (0x00000004)

driver.sendKeyEvent(4);

// 4.4 To stop automating in the web view context we can simply call the context again with id NATIVE\_APP.

for(String context : availableContexts) {

if(context.contains("NATIVE")){

System.out.println("We are Back to " + context);

driver.context(context);

if (driver.findElement(title).getText().equals("WebViewApp"))

System.out.println("Context Switched");

}

}

}

@AfterTest(alwaysRun= true)

public void tearDown(){

driver.quit();

System.out.println("tearDown() :: driver.quit() executed");

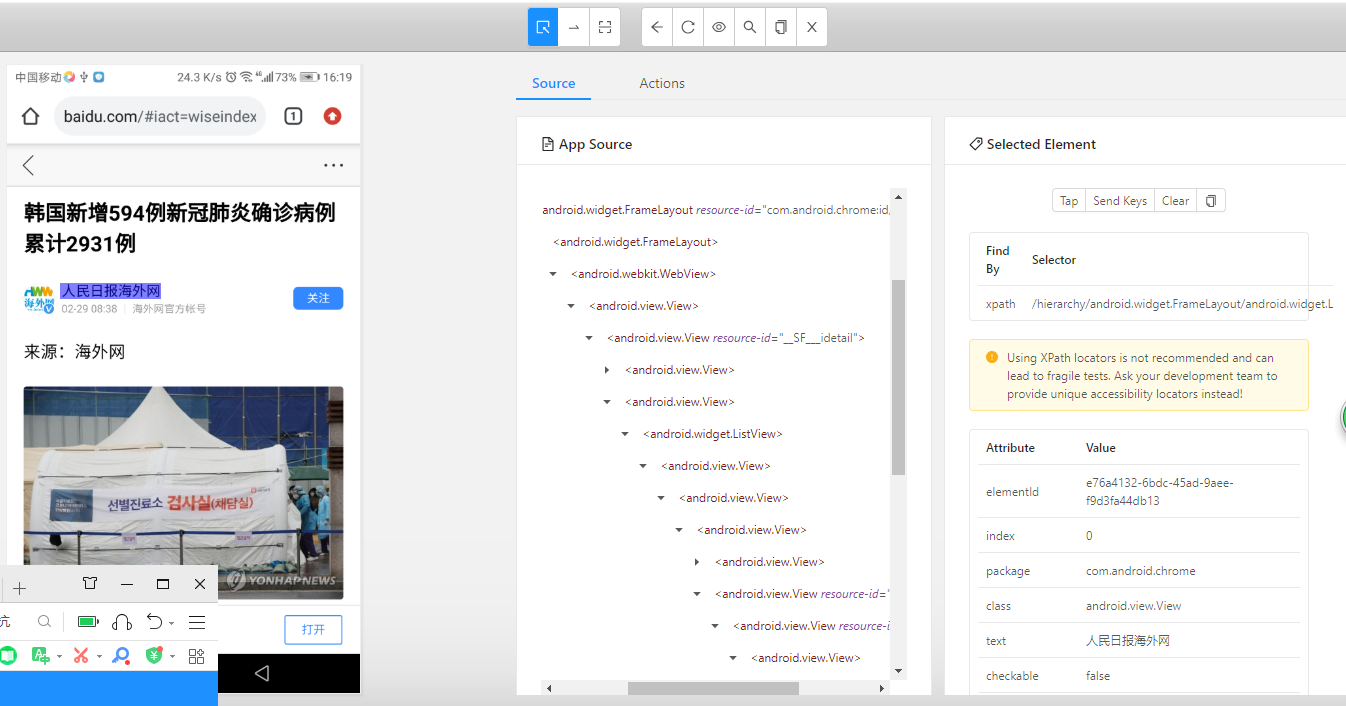
}

} // end of class

appium需要配置platformname，platformversion，devicename才能连接app

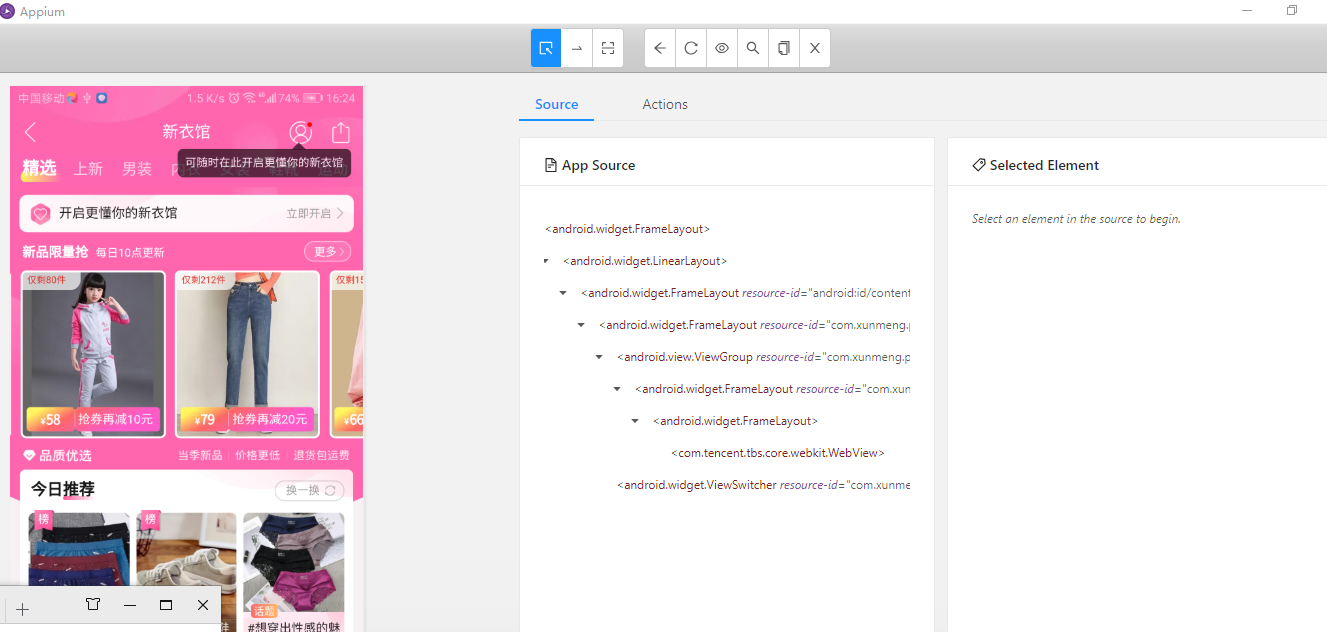
Appium可以定位xpath及元素信息

Xpath=/hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.view.ViewGroup/android.widget.FrameLayout[1]/android.widget.FrameLayout[1]/android.webkit.WebView/android.view.View/android.view.View/android.view.View[2]/android.widget.ListView/android.view.View/android.view.View/android.view.View/android.view.View[2]/android.view.View[1]/android.view.View[1]/android.view.View[2]/android.view.View[1]/android.view.View[2]/android.view.View



未开启talkback时截图

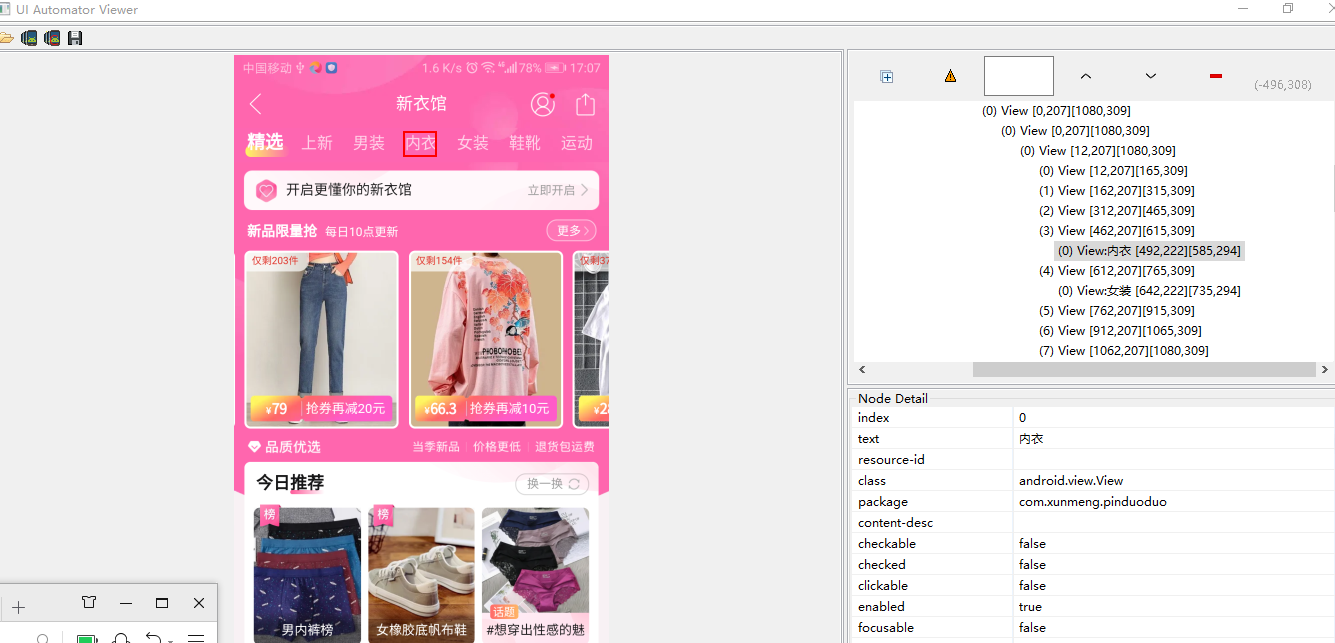
App里包含webview

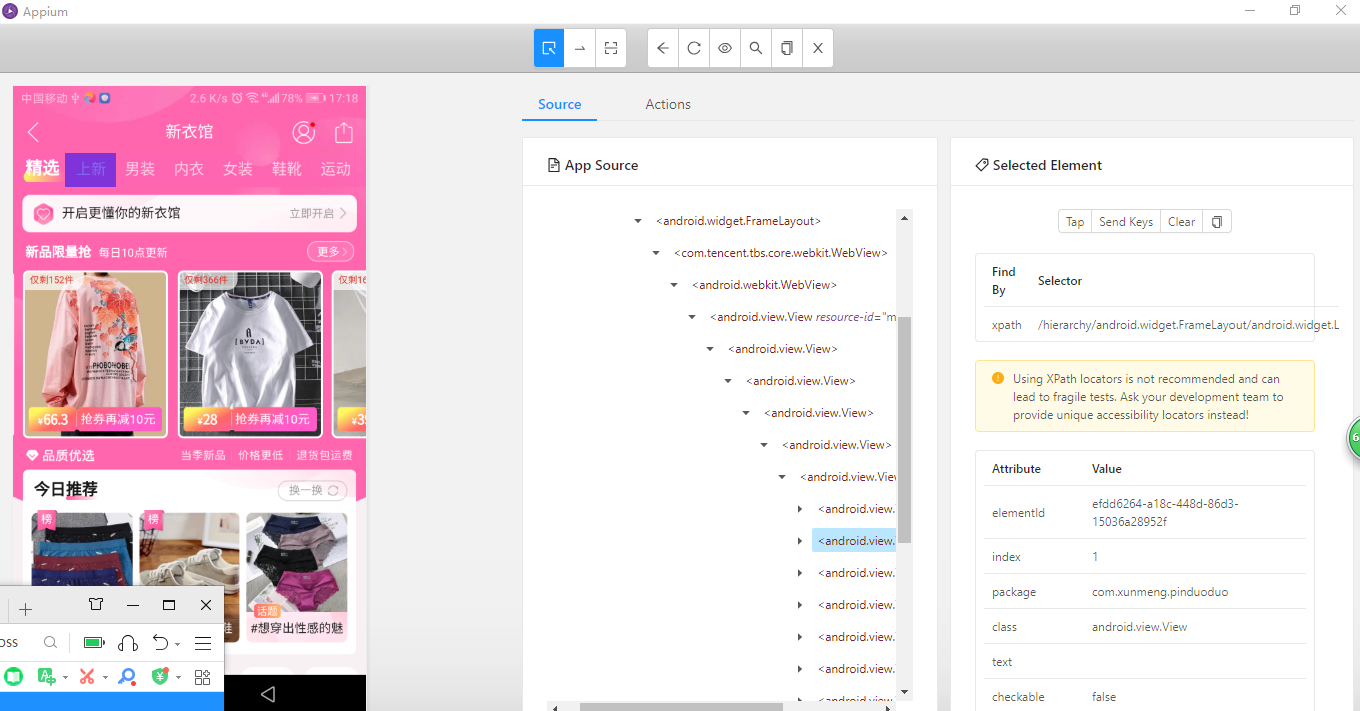


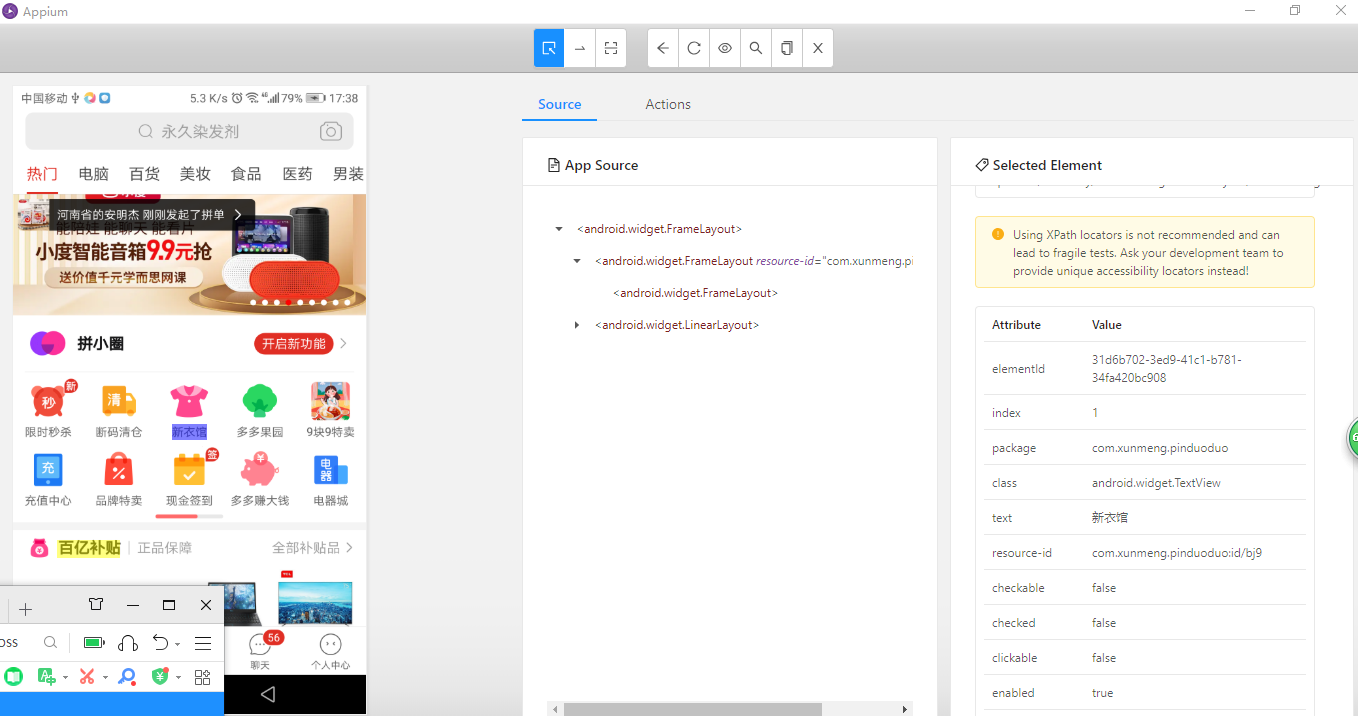
开启talback

Webview很少有resourceid属性，其他属性基本也没有，只能通过xpath定位元素

拼多多，webview页面







案例；拼多多

name=com.xunmeng.pinduoduo/com.xunmeng.pinduoduo.ui.activity.HomeActivity)

首页按钮元素：

图片：Resourceid：com.xunmeng.pinduoduo:id/ail 下面几个按钮都是相同id

Xapth：

/hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.RelativeLayout/android.widget.LinearLayout/android.widget.RelativeLayout[1]/android.widget.ImageView

文字：

Resouceid：com.xunmeng.pinduoduo:id/d8q，有很多个

Text：首页

Xpath：

/hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.RelativeLayout/android.widget.LinearLayout/android.widget.RelativeLayout[1]/android.widget.TextView

新衣馆元素：

Resourceid：com.xunmeng.pinduoduo:id/ah9

Xpath：/hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.RelativeLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget.LinearLayout/android.support.v4.view.ViewPager/android.widget.FrameLayout/android.support.v7.widget.RecyclerView/android.widget.LinearLayout/android.widget.FrameLayout/android.support.v7.widget.RecyclerView/android.widget.RelativeLayout[3]/android.widget.ImageView

文字

Text：新衣馆

Resourceid：com.xunmeng.pinduoduo:id/bj9

Xpath：

/hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.RelativeLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.widget.LinearLayout/android.support.v4.view.ViewPager/android.widget.FrameLayout/android.support.v7.widget.RecyclerView/android.widget.LinearLayout/android.widget.FrameLayout/android.support.v7.widget.RecyclerView/android.widget.RelativeLayout[3]/android.widget.TextView

内衣元素：

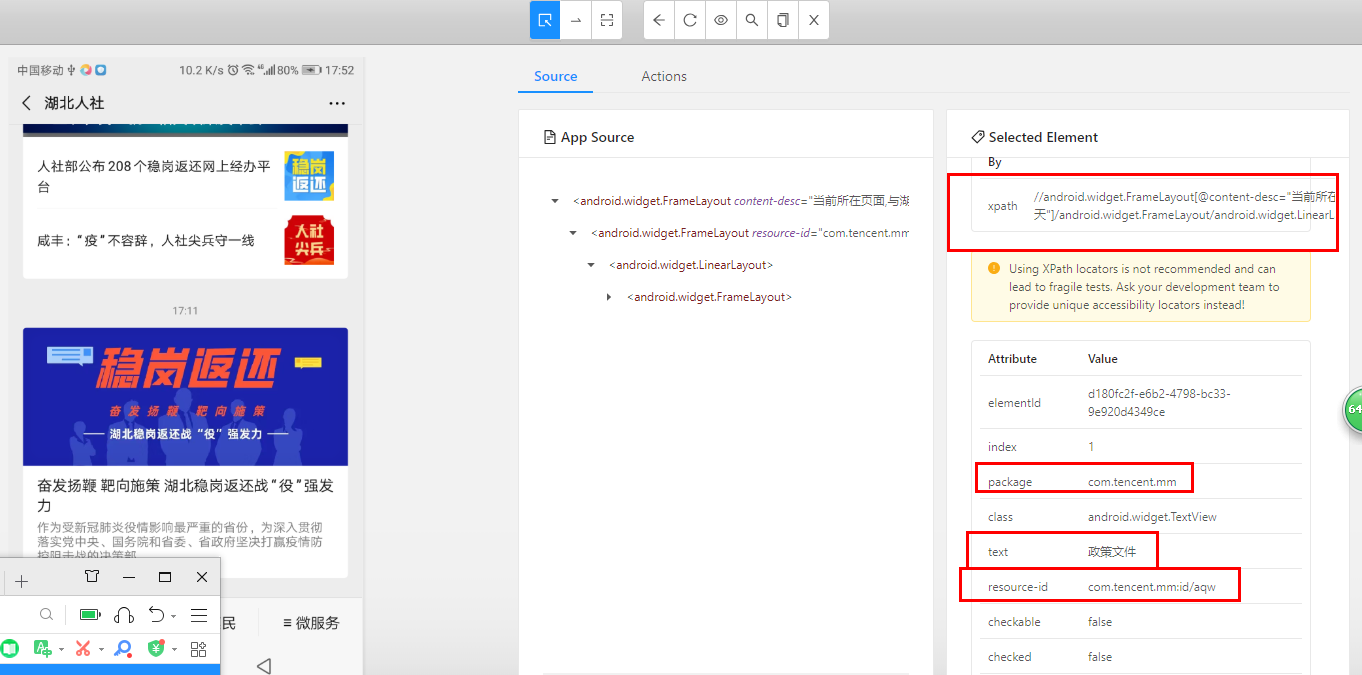
Text：内衣

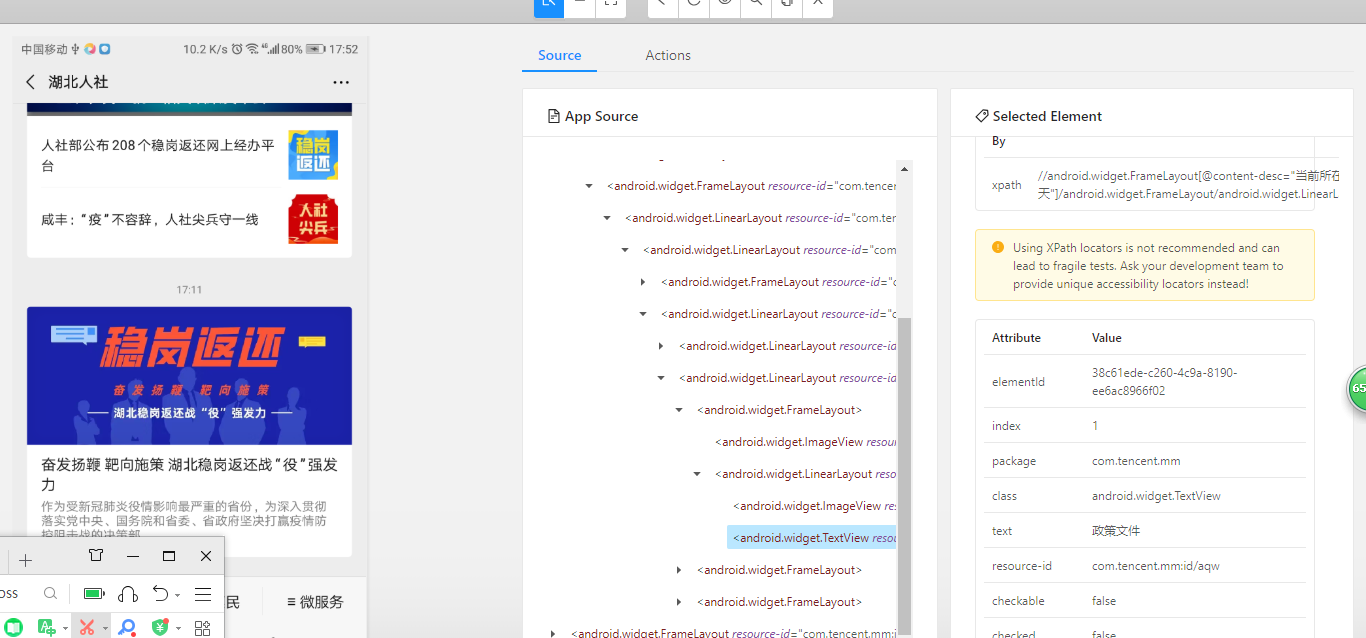
Xpath：

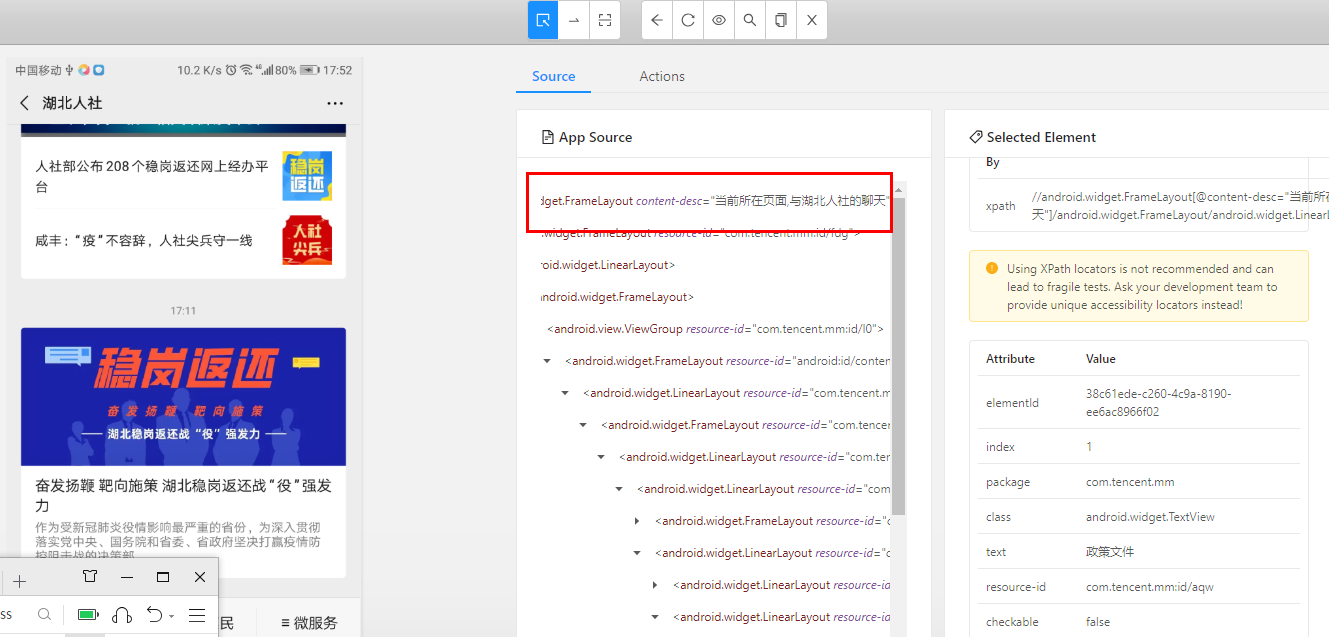
/hierarchy/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.FrameLayout/android.view.ViewGroup/android.widget.FrameLayout/android.widget.FrameLayout/com.tencent.tbs.core.webkit.WebView/android.webkit.WebView/android.view.View[1]/android.view.View[1]/android.view.View/android.view.View/android.view.View/android.view.View/android.view.View[4]/android.view.View

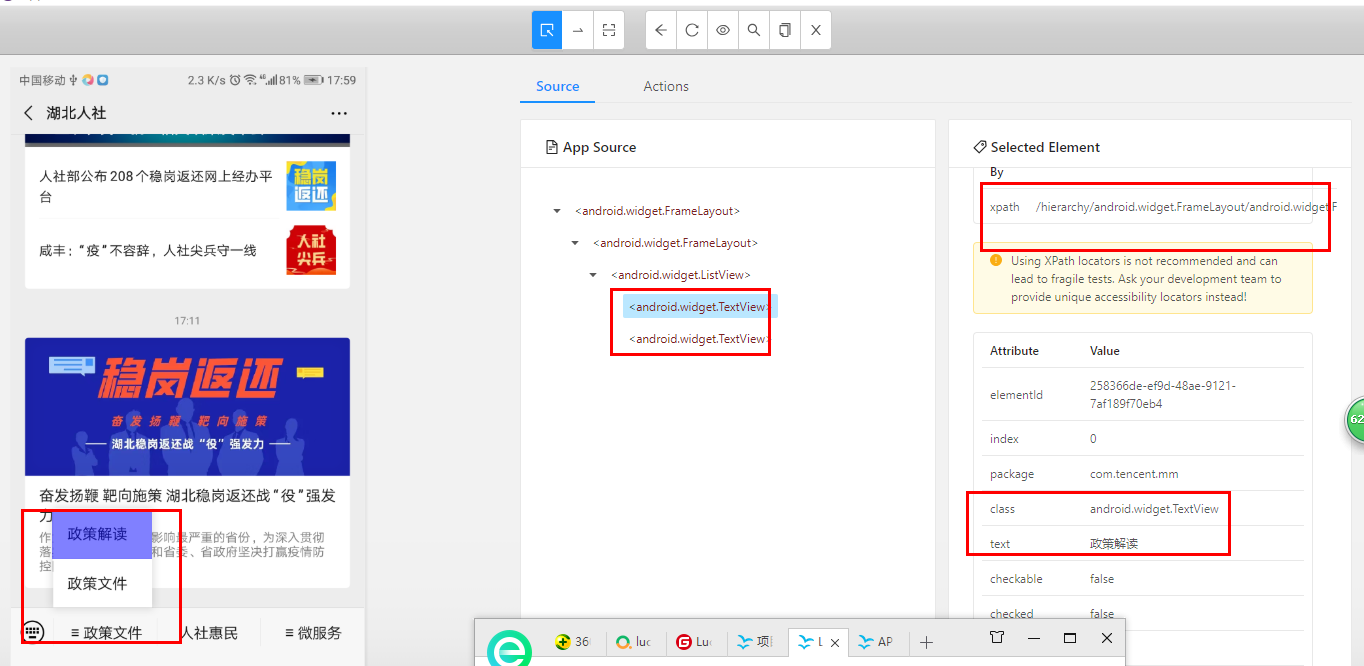
微信公众号

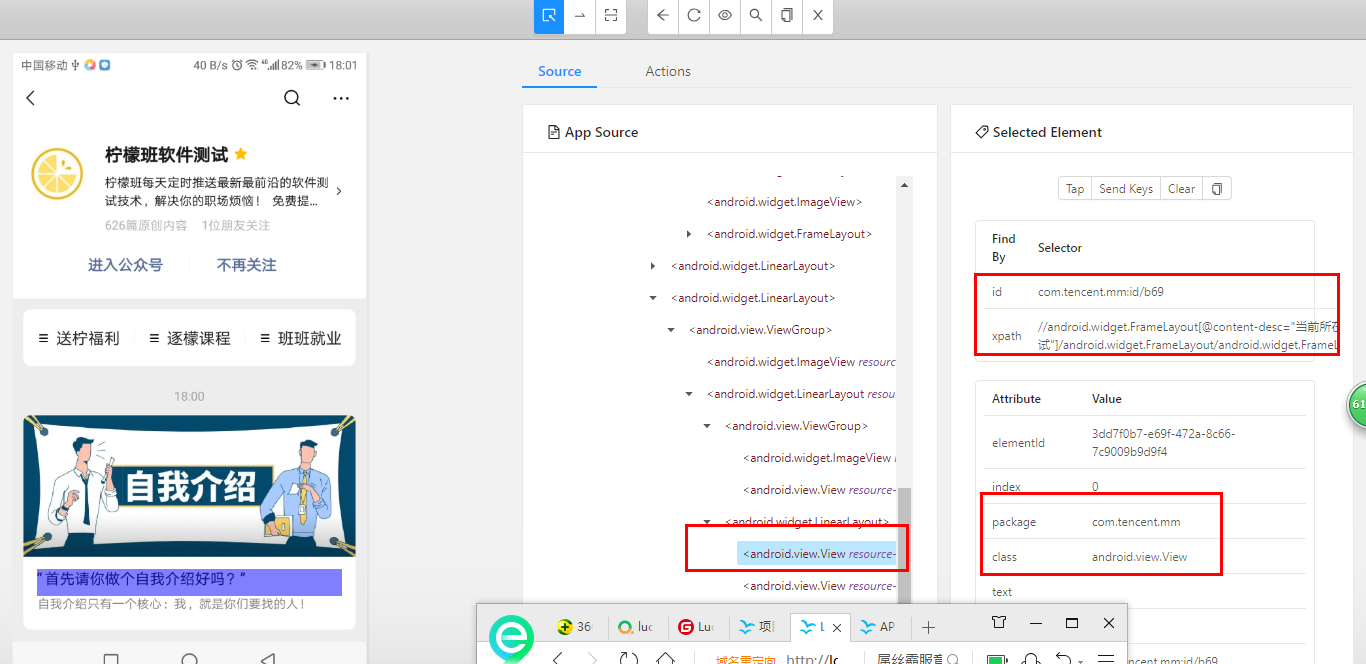
湖北人社











政策文件

Resourceid：com.tencent.mm:id/aqw

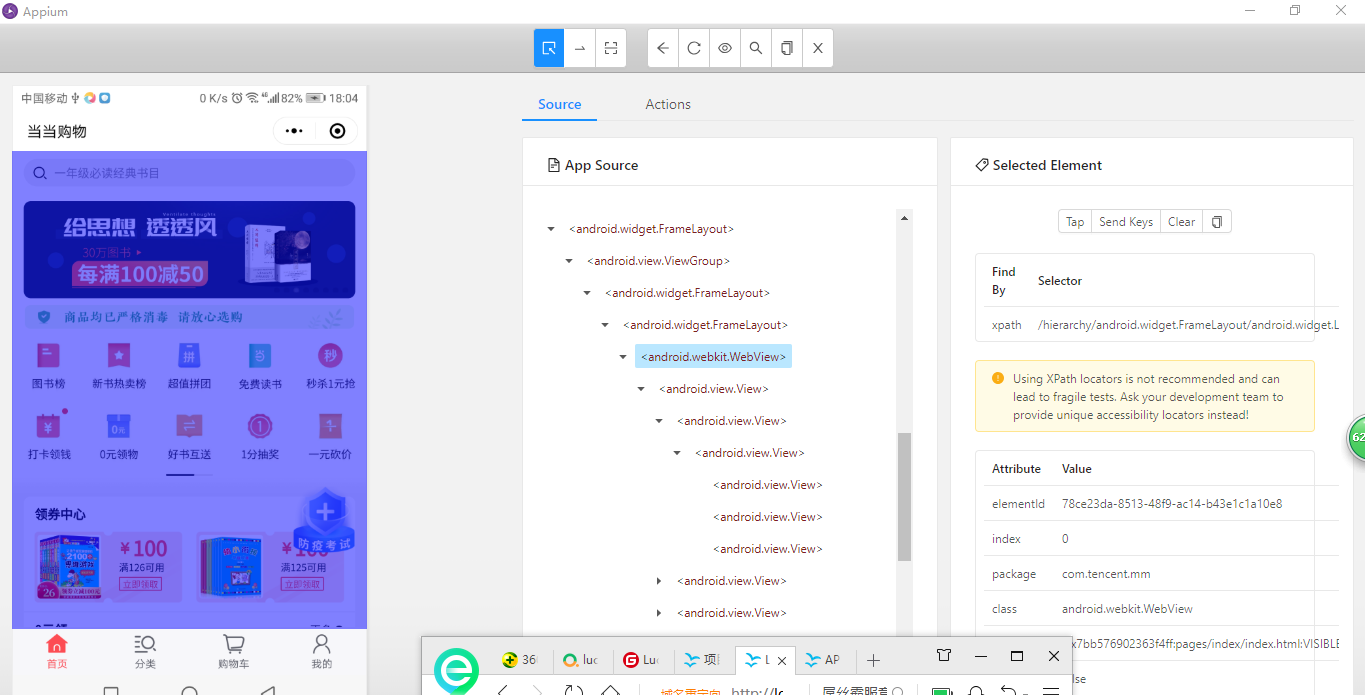
Xpath：

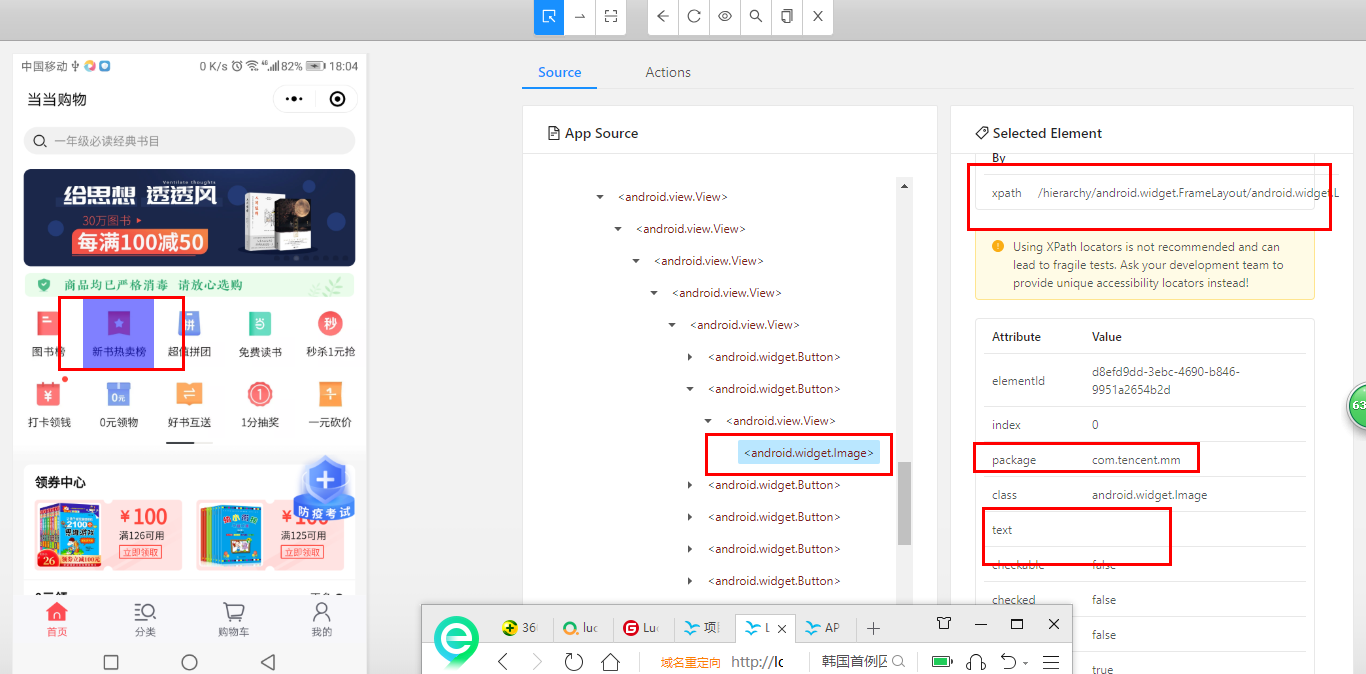
//android.widget.FrameLayout[@content-desc="当前所在页面,与湖北人社的聊天"]/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.FrameLayout/android.view.ViewGroup/android.widget.FrameLayout[1]/android.widget.LinearLayout/android.widget.FrameLayout/android.widget.LinearLayout/android.widget.LinearLayout/android.widget.LinearLayout/android.widget.LinearLayout[2]/android.widget.FrameLayout[1]/android.widget.LinearLayout/android.widget.TextView

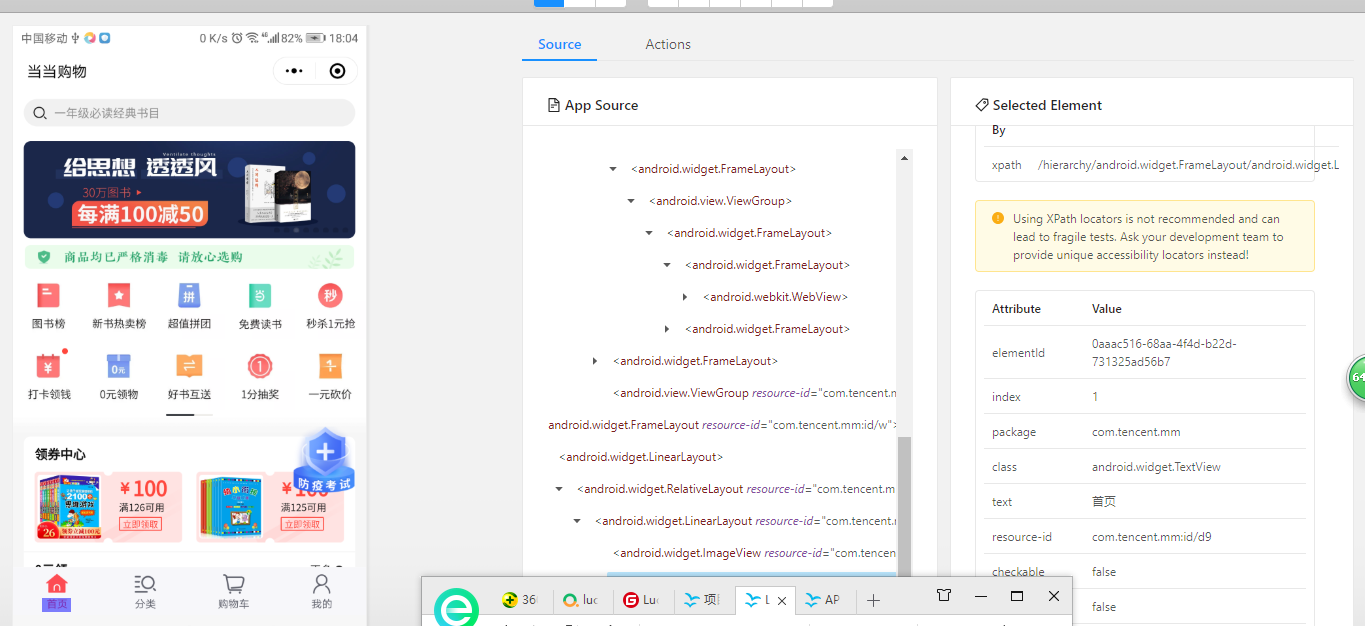
微信小程序

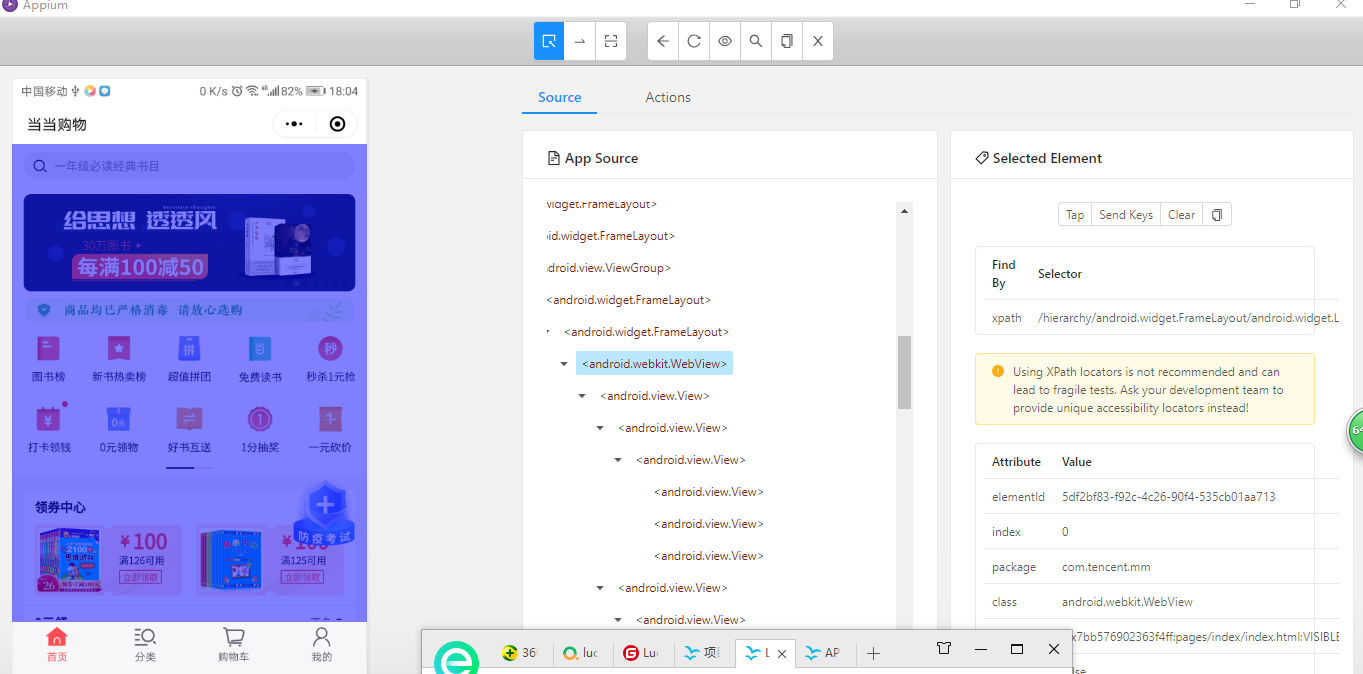
Webview

当当网









手机访问web网页，

Github

Uc-devtools inspect

Google devtools inspect

都可以

微信小程序，公众号

前提：在打开的界面中选择信息->勾选是否打开TBS内核Inspector调试功能

支持腾讯x5内核

Google devtools，需翻墙

Uc-devtools 不翻墙