

# 微信小程序测试

## 小程序特点

## 官网文档

<https://developers.weixin.qq.com/miniprogram/dev/framework/quickstart/#%E4%BD%93%E9%AA%8C%E5%B0%8F%E7%A8%8B%E5%BA%8F>

### 小程序与普通网页开发的区别

小程序的主要开发语言是 JavaScript，小程序的开发同普通的网页开发相比有很大的相似性。对于前端开发者而言，从网页开发迁移到小程序的开发成本并不高，但是二者还是有些许区别的。

网页开发渲染线程和脚本线程是互斥的，这也是为什么长时间的脚本运行可能会导致页面失去响应，而在小程序中，二者是分开，分别运行在不同的线程中。网页开发者可以使用到各种浏览器暴露出来的 DOM API，进行 DOM 选中和操作。而如上文所述，小程序的逻辑层和渲染层是分开的，逻辑层运行在 JSCore 中，并没有一个完整浏览器对象，因而缺少相关的 DOM API 和 BOM API。这一区别导致了前端开发非常熟悉的一些库，例如 jQuery、Zepto 等，在小程序中是无法运行的。同时 JSCore 的环境同 NodeJS 环境也是不尽相同，所以一些 NPM 的包在小程序中也是无法运行的。

网页开发者需要面对的环境是各式各样的浏览器，PC 端需要面对 IE、Chrome、QQ 浏览器等，在移动端需要面对 Safari、Chrome 以及 iOS、Android 系统中的各式 WebView。而小程序开发过程中需要面对的是两大操作系统 iOS 和 Android 的微信客户端，以及用于辅助开发的小程序开发者工具，小程序中三大运行环境也是有所区别的，如表 1-1 所示。

表 1-1 小程序的运行环境

运行环境	逻辑层	渲染层
iOS	JavaScriptCore	WKWebView
安卓	V8	chromium 定制内核
小程序开发者工具	NWJS	Chrome WebView

网页开发者在开发网页的时候，只需要使用到浏览器，并且搭配上一些辅助工具或者编辑器即可。小程序的开发则有所不同，需要经过申请小程序帐号、安装小程序开发者工具、配置项目等等过程方可完成。

# 元素获取

## UC 开发者工具

AOSP on Shama #127.0.0.1:62001


com.tencent.mm (74.0.3729.136)


没有检测到 webview

com.tencent.mm (74.0.3729.136)

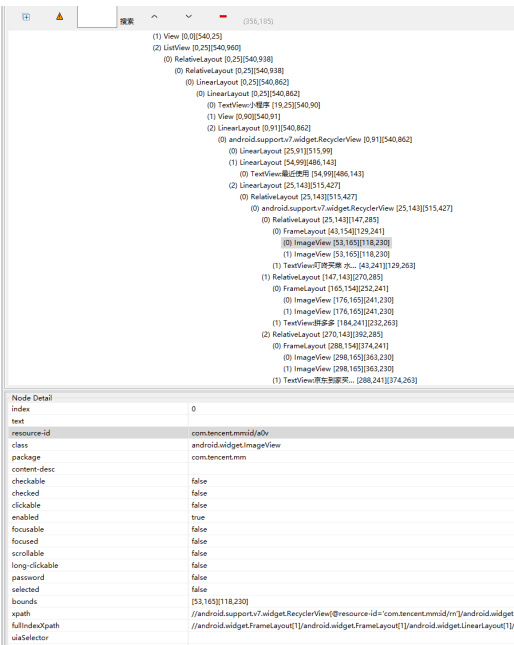
没有检测到 webview

com.tencent.mm (74.0.3729.136)

 about:blank about:blank  
[inspect](#)

 wx1e113254eda17715:pages/mainPackage/home/home.html:VISIBLE about:blank  
[inspect](#)

uiautomatorviewer



```
['NATIVE_APP', 'WEBVIEW_com.tencent.mm:appbrand0']
Traceback (most recent call last):
```



小程序：知乎

微信账号：建议使用小号

获取小程序进程号

打开被测小程序

adb shell dumpsys activity top | findstr ACTIVITY

```
C:\Users\ThinkPad>adb shell dumpsys activity top | findstr ACTIVITY
ACTIVITY com.tencent.mm/.plugin.appbrand.ui.AppBrandUI 2612997c pid=5072
```

获取Android进程

adb shell ps 5072

```
C:\Users\ThinkPad>adb shell ps 5072
USER      PID    PPID  VSIZE  RSS      WCHAN    PC         NAME
u0_a67    5072   1516   1905388 176984  ffffffff b747c0f5 S  com.tencent.mm:appbrand0
```

chromeOptions:

Appium官方预留了chromeOptions选项，在capability里可以添加这个参数来自定义配置ChromeDriver 会话的选项

<b>chromeOptions</b>	Allows passing <b>chromeOptions</b> capability for ChromeDriver. For more information see <b>chromeOptions</b>	<b>chromeOptions</b> : {args: ['--disable-popup-blocking']}
----------------------	--	---

**注意：**微信小程序，用模拟器可能是搞不定的，模拟器确实很容易闪退，夜神模拟器最新的版本是7.0的，用真机测试比较靠谱

代码实现：

```
from appium import webdriver
import time

from selenium.webdriver.common.by import By
from selenium.webdriver.support.wait import WebDriverWait

class Test:
    def setup(self):

        desired_caps = {}
        desired_caps['platformName'] = 'Android'
        desired_caps['platformVersion'] = '5.1.1'
        desired_caps['deviceName'] = 'demo'
        desired_caps['appPackage'] = 'com.tencent.mm'
        desired_caps['appActivity'] = '.ui.LauncherUI'
```

```

        desired_caps['unicodekeyboard'] = True
        desired_caps['noReset'] = True
        desired_caps['chromeOptions']=
{'androidProcess':'com.tencent.mm:appbrand0'}
        self.driver = webdriver.Remote('http://127.0.0.1:4723/wd/hub',
desired_capabilities=desired_caps)
        self.driver.implicitly_wait(50)
        time.sleep(20)

    def find(self,by,loc):
        return WebDriverWait(self.driver,timeout=50).until(lambda x
:self.driver.find_element(by,loc))

    def test_case01(self):
        time.sleep(20)
        size = self.driver.get_window_size()
        self.driver.swipe(size['width'] * 0.5, size['height'] * 0.25,
size['width'] * 0.75, size['height'] * 0.75, 3000)
        time.sleep(20)
        self.find(By.ID,"com.tencent.mm:id/a0v").click()
        time.sleep(30)
        print(self.driver.contexts)
        time.sleep(10)
        # 定位搜索框
        self.find(By.XPATH,"/html/body/wx-view/wx-nav-bar/wx-view[2]/wx-
view[2]/wx-search-bar/wx-view/wx-view").click()
        # 点击我的
        self.find(By.XPATH,'//*[@text="我的"]').click()

```

微信历史版本:

[https://weixin.qq.com/cgi-bin/readtemplate?t=weixin\\_faq\\_list\\_6](https://weixin.qq.com/cgi-bin/readtemplate?t=weixin_faq_list_6)