# OPPO A92 微信键盘兼容性测试

### 自动化测试环境

[**Appium+Python自动化 1 环境搭建（适用windows系统-Android移动端自动化）**](https://www.cnblogs.com/baby0511/p/10430127.html)

Python+Appium+Unittest+Jenkins

链接：https://pan.baidu.com/s/1WRA4Imip8lgmggRxiUb5Vg

提取码：abcd

##### adb 的工作原理

启动一个 adb 客户端时，此客户端首先检查是否有已运行的 adb 服务器进程。如果没有，它将启动服务器进程。当服务器启动时，它与本地 TCP 端口 5037 绑定，并侦听从 adb 客户端发送的命令—所有 adb 客户端均使用端口 5037 与 adb 服务器通信。

###### “获取包名和 Activity 的方法”，用于自动化测试时启动 APP

1. 方法一： pm list package 查看包名

adb shell pm list package -f

将获取手机内所有 apk 对应的包名和路径

2. 方法二：dumpsys window（本人最常用）

先打开需要查找的 apk 后，再敲入以上命令

adb shell dumpsys window w | findstr \/ | findstr name=

或

adb shell dumpsys window |findstr mCurrent

4. 方法三：dumpsys activity

先启动需要的查找的 apk，再敲入以上命令

adb shell dumpsys activity | grep mFocusedActivity

##### 安装Python3.6+

安装依赖环境：pip3 install -r requirements.txt

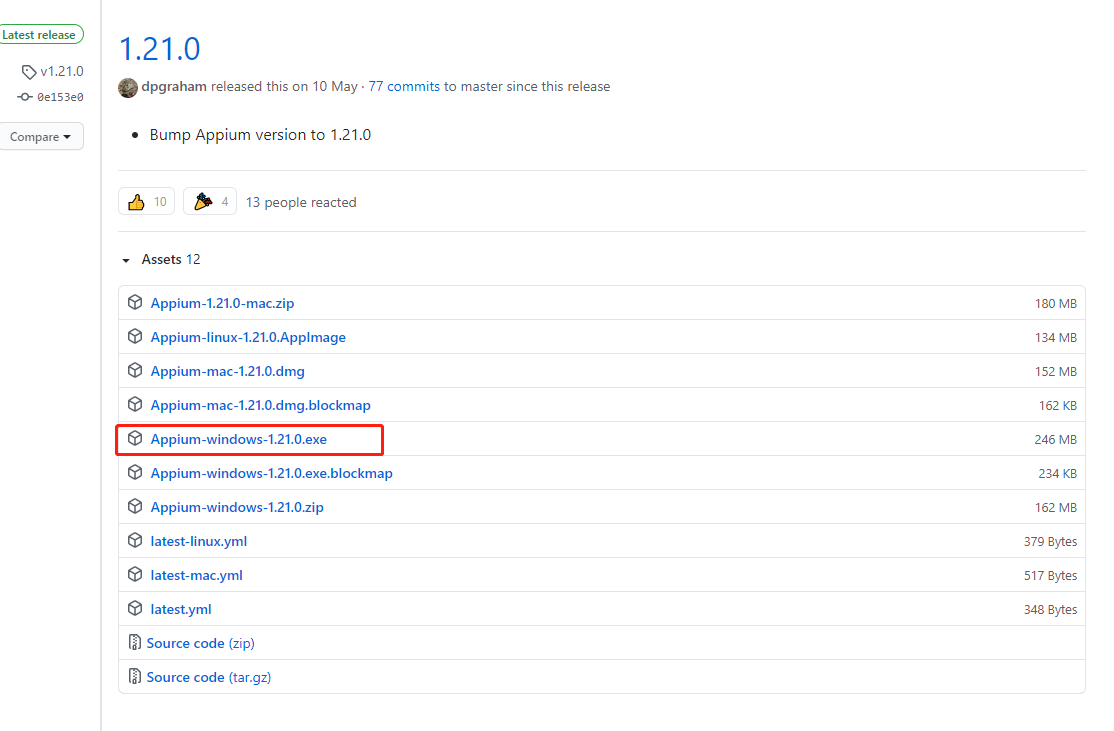
[Python 自动生成环境依赖包 requirements\_fenglepeng的博客-CSDN博客](https://blog.csdn.net/fenglepeng/article/details/110797565)

##### 安装Appium Server

网盘链接 下载安装：Appium-windows-1.15.1.exe

Appium Desktop官方下载：

<https://github.com/appium/appium-desktop/releases/tag/v1.21.0>



##### 安装JDK

网盘链接 下载安装：jdk-8u211-windows-x64.exe

安装好之后，还需要添加一个环境变量 JAVA\_HOME ，指定值为jdk安装目录，比如：

JAVA\_HOME D:\JDK

##### 安装 Android SDK

网盘链接 下载最新的 Android SDK文件包：androidsdk.zip ，并且解压。

解压完成后，需要配置一下添加一个环境变量 ANDROID\_HOME ，设置值为sdk包解压目录，比如 C:\Users\v\_zengmliu\AppData\Local\Android\Sdk

##### 连接手机

上述的软件环境都准备好以后，要自动化手机APP，需要：

* 在运行程序的电脑上，用USB线连接上安卓手机
* 进入 手机设置 -> 关于手机 ，不断点击 版本号 菜单（7次以上），
* 退出到上级菜单，在开发者模式中，启动USB调试

连接好以后，打开命令行窗口，执行 adb devices -l 命令来列出连接在电脑上的安卓设备。

如果输出 类似如下的内容：

List of devices attached

89LJQWROR47TCYKJ device product:PDKM00 model:PDKM00 device:OP4ABB transport\_id:1

表示电脑上可以查看到连接的设备，就可以运行自动化程序了。

##### Jenkins集成

[Jenkins自动化部署入门详细教程](https://www.cnblogs.com/wfd360/p/11314697.html)

打开浏览器进入链接 [http://localhost:8080](http://localhost:8080/)

[Jenkins集成appium自动化测试(Windows篇)](https://blog.csdn.net/qq_42434318/article/details/109403869)

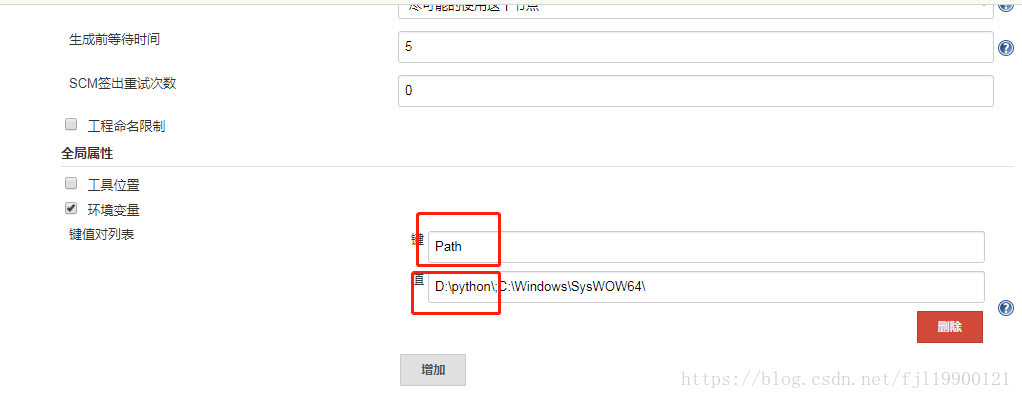
##### jenkins坑

1. 无法拉取远程仓库的工程

排查：检查项目URL是否正确，检查仓库地址是否正确，检查GITHUB用户名和密码是否正确，检查git-bash.exe路径是否正确（本次错误原因）



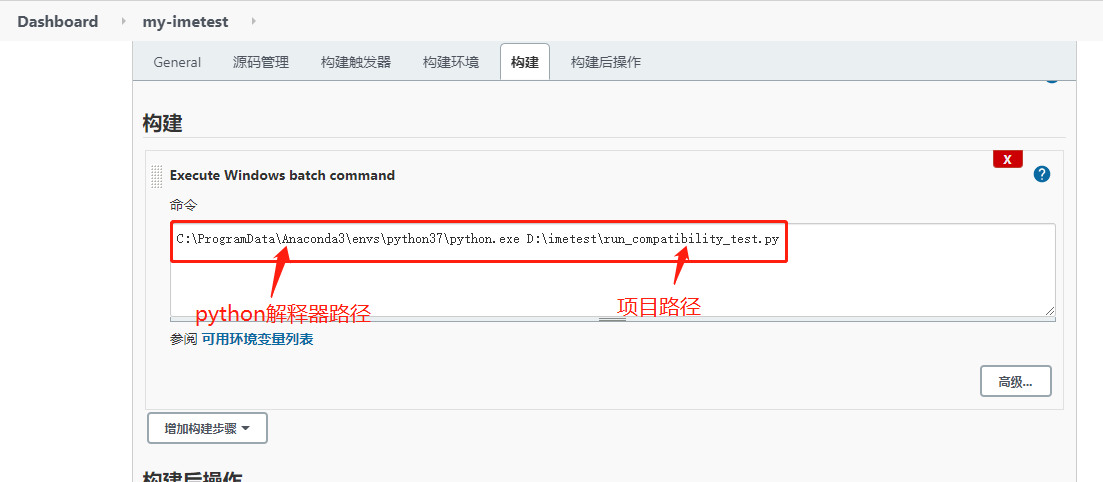
**python不是内部命令**，就需要配置环境变量:



或者指明路径：（我采取该策略）

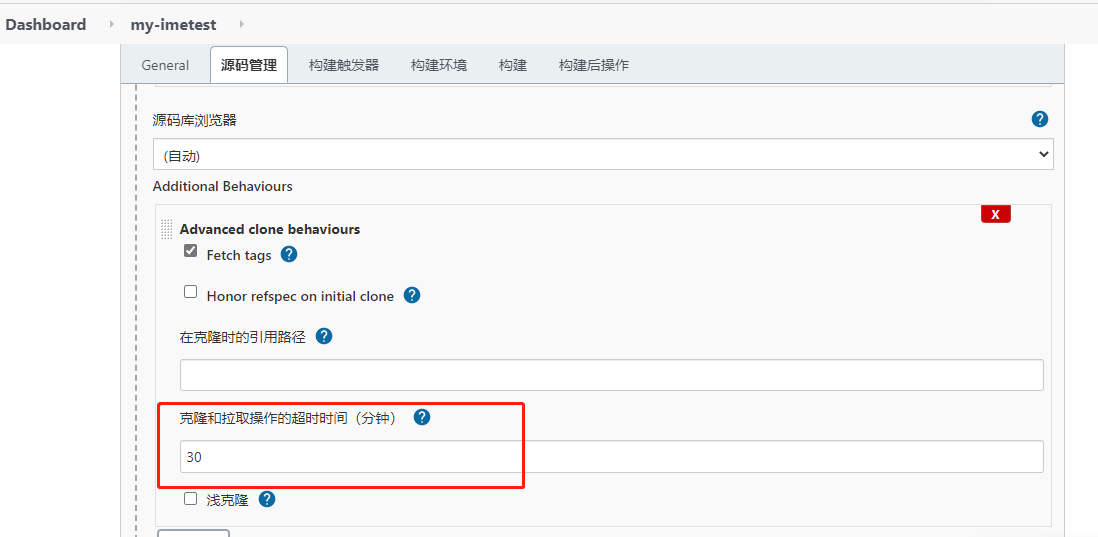
错因：python \*\*.py

解决方法如下图：



1. 项目太大导致git拉取工程失败

解决方法：



##### git删除远程文件夹或文件的方法

项目开发初期由于.gitignore 文件配置不正确很有可能导致某些不需要的目录上传到 git 远程仓库上了，这样会导致每个开发者提交的时候这些文件每次都会不同。除了一开始提交的时候注意配置好 .gitignore 文件外，我们也需要了解下出现这种问题后的解决办法。

具体操作步骤如下：

1. 预览将要删除的文件
2. git rm -r -n --cached 文件/文件夹名称
3. 加上 -n 这个参数，执行命令时，是不会删除任何文件，而是展示此命令要删除的文件列表预览。
4. 确定无误后删除文件
5. git rm -r --cached 文件/文件夹名称
6. 提交到本地并推送到远程服务器
7. git commit -m "提交说明"
8. git push origin master
9. 修改本地 .gitignore 文件 并提交
10. git commit -m "提交说明"

git push origin master

### 测试内容

微信键盘在Oppo A92手机各app上的兼容性测试

### 测试范围

1. **app：**微信（搜索框，朋友圈，聊天框）、QQ（账号框，密码框）、抖音、爱奇艺、支付宝、微博、哔哩哔哩、便签、美团、全民K歌、淘宝、百度、今日头条、网易云（共14个app）

**②键盘功能：**九宫格（中文，英文，数字，标点符号，长按操作）、全键盘（中文，英文，数字，标点符号，表情符号，长按操作）

### 元素查看工具

Ui Automator Viewer、Appium的Inspector、Airtest

### 元素定位方式

1. **键盘元素定位：**

坐标（因为工具定位微信键盘时，**只能定位某一片区域，具体的元素无法定位**）

1. **App元素定位：**

**appium定位app上的元素方式 在前面文章中有介绍，（通过UIautomator工具查看app元素）**

**appium定位app上的元素，可以通过id，name，class这些属性定位到**

**1、id 定位，是需要找到元素的 resource-id属性**

**格式：driver.find\_element\_by\_id("resource-id属性值")**

**2、name 定位，是要找到元素的 text属性**

**格式：driver.find\_element\_by\_name("text属性值")**

**3、class 定位，是要找到元素的class属性**

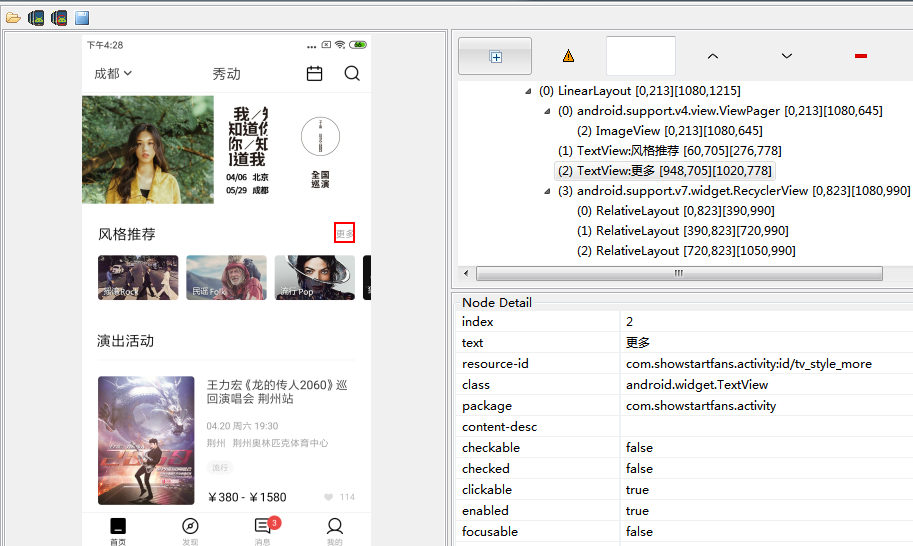
**格式：driver.find\_element\_by\_class\_name("class属性值")**

**4、accessibility\_id 定位，是要找到元素的content-desc属性**

**格式：driver.find\_element\_by\_accessibility\_id("content-desc属性值")**

**5、xpath 定位**

**这儿以秀动app为例，定位风格推荐的“更多”按钮**



**①如果元素text是唯一的，可以通过text文本定位**

**格式： //\*[@text='text文本属性值']**

**eg：driver.find\_element\_by\_xpath("//\*[@text='更多']").click()**

**②如果元素id是唯一的，可以通过id属性定位**

**格式：//\*[@resource-id='resource-id属性值']**

**eg：driver.find\_element\_by\_xpath("//\*[@resource-id='com.showstartfans.activity:id/tv\_style\_more']").click()**

**可以联合@resource-id属性和@text文本属性定位**

**driver.find\_element\_by\_xpath("//\*[@resource-id='com.showstartfans.activity:id/tv\_style\_more'][@text='更多']").click()**

**③如果class属性唯一，也可以通过class属性定位（有两种方法）**

**a、第一种： 格式： //class属性值**

**eg：driver.find\_element\_by\_xpath("//android.widget.TextView").click()**

**b、第二种： 格式： //\*[@class='class属性值']**

**eg：driver.find\_element\_by\_xpath("//\*[@class='android.widget.TextView']").click()**

**④通过content-desc属性定位**

**格式： //\*[@content-desc='desc的属性值']**

**eg：driver.find\_element\_by\_xpath("//\*[@content-desc='XXX']").click()**

**⑤contains 模糊定位**

**contains是模糊匹配的定位方法，对于一个元素的id或者text不是固定的，但有一部分是固定的，这种就可以模糊匹配**

**格式：//\*[contains(@content-desc, ‘属性值’)]**

**//\*[contains(@resource-id, ‘属性值’)]**

**//\*[contains(@clsss, ‘属性值’)]**

**⑥组合定位**

**如果一个元素有2个属性，通过xpath也可以同时匹配2个属性,text, resource-id,class ,index,content-desc这些属性都能任意组合定位**



**注意事项：**

1、如果元素的resource-id，text，class，content-desc属性值为空，那么就不能通过该相关方式定位

2、一般一个页面上的class属性不唯一，元素不唯一的话定位会报错了，非必要条件下，建议不使用class属性定位

3、appium1.5以下老的版本是可以通过name定位的，新版本从1.5以后都不支持name定位了

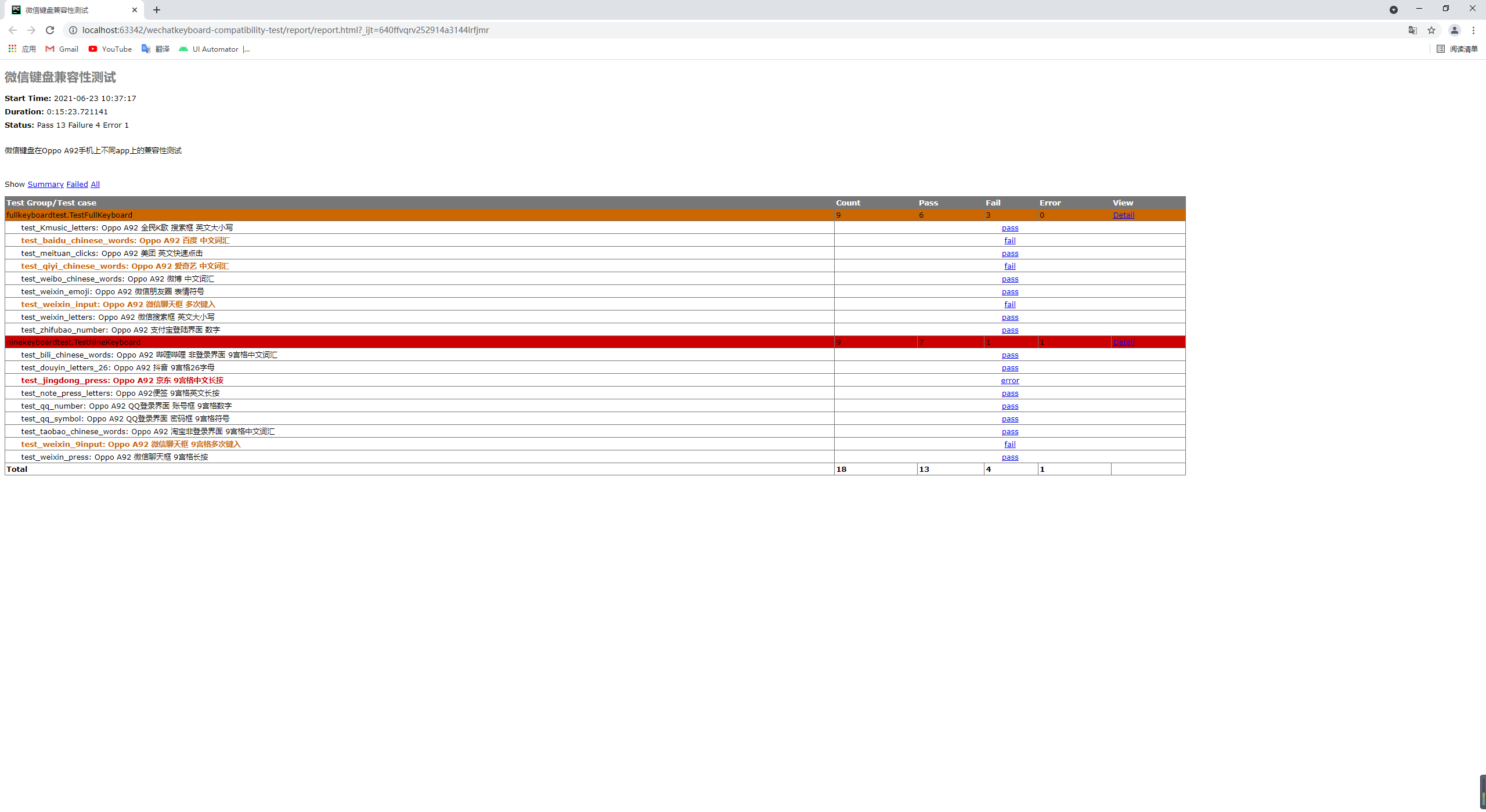
### 持续集成（Jenkins）

每20分钟构建一次，一次执行时间大约在12分钟左右

### 测试报告

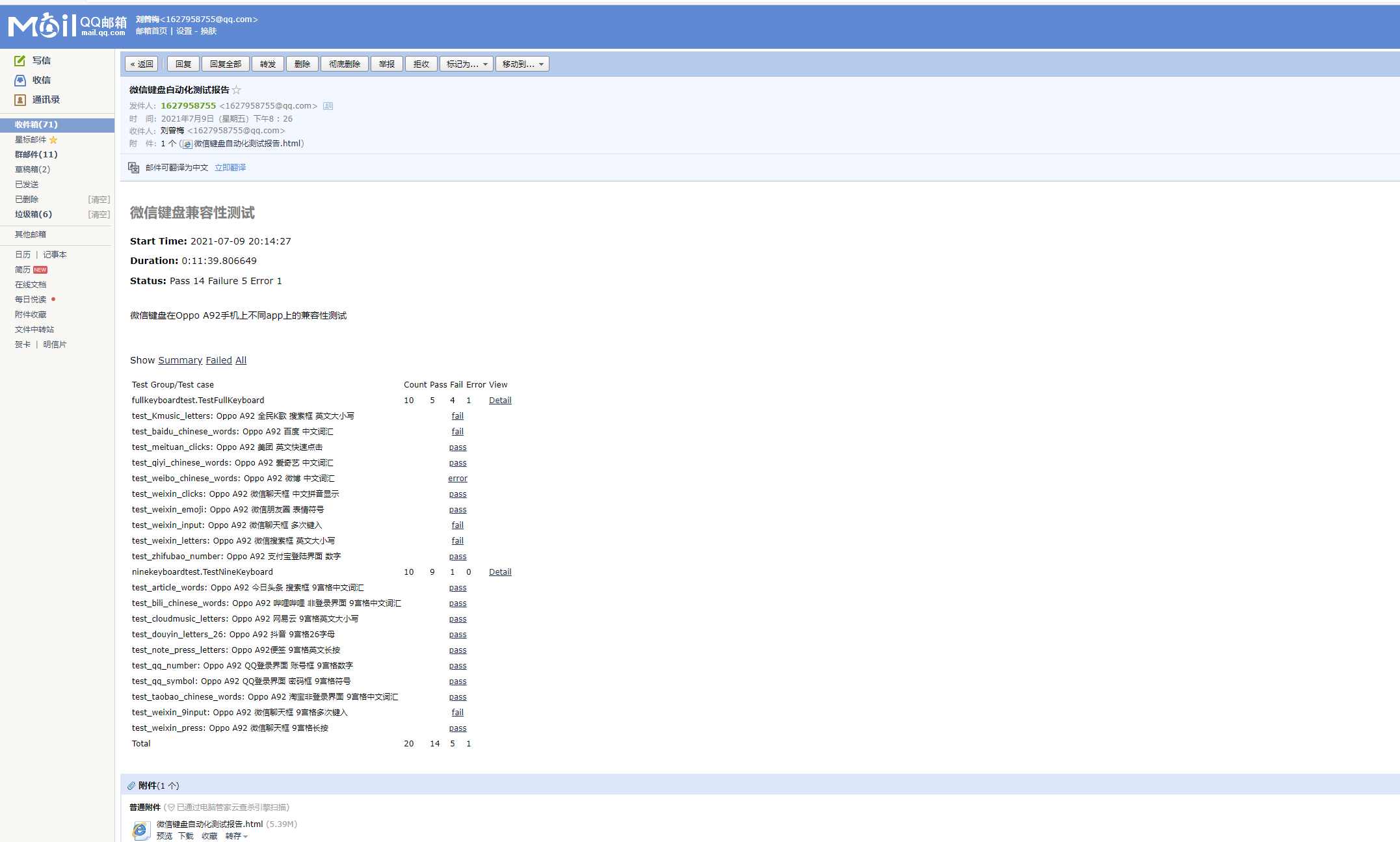
每执行完一次测试，生成测试报告

HTML格式



### 发送邮件

每执行完一次测试，发送邮件



### 测试用例

前提条件：执行用例前均切换到微信输入法（自动切换）

Class1：全键盘

Case1:

描述：测试全键盘英文大小写在全民K歌app搜索框的可用性

步骤：①打开全民K歌app，点击搜索框

②切换为英文模式，点击“大写”按钮，依次点击‘A’ ‘n’ ‘g’ ‘e’ ‘l’ ‘s’字母

③获取搜索框内容

④断言

预期返回值：Angels

实际返回值：Angels

Case2:

描述：测试全键盘英文快速点击在美团app搜索框的可用性

步骤：①打开美团app，点击搜索框

②切换为英文模式快速依次点击‘y’ ‘t’ ‘d’ ‘k’ ‘c’ 字母

③获取搜索框内容

④断言

预期返回值：ytdkc

实际返回值：ytdkc

Case3:

描述：测试全键盘中文待选框词汇在爱奇艺app搜索框的可用性

步骤：①打开爱奇艺app，点击搜索框

②依次点击‘j’ ‘u’ ‘e’ ‘x’ ‘i’ ‘n’ ‘g’ ‘n’ ‘i’ ‘a’ ‘n’ ‘d’ ‘a’ ‘I’ 拼音

③点击待选框第一个词汇

④获取搜索框内容

⑤断言

预期返回值：觉醒年代

实际返回值：觉醒年代

Case4:

描述：测试全键盘中文待选框词汇在微博app搜索框的可用性

步骤：①打开微博app，点击搜索框

②依次点击‘w’ ‘e’ ‘i’ ‘x’ ‘i’ ‘n’ ‘j’ ‘i’ ‘a’ ‘p’ ‘a’ ‘n’ 拼音

③点击待选框第一个词汇

④获取搜索框内容

⑤断言

预期返回值：微信键盘

实际返回值：微信键盘

Case5:

描述：测试全键盘表情符号在微信app朋友圈的可用性

步骤：①打开全民K歌app，点击搜索框

②切换为符号模式，点击“表情符号”，依次点击表情

④保存文本框截图

预期返回值：

实际返回值：

Case6:

描述：测试全键盘英文大小写在微信app搜索框的可用性

步骤：①打开微信app，点击搜索框

②切换为英文模式，点击“大写”按钮，依次输入‘X’ ‘i’ ‘a’ ‘o’ ‘L’ ‘i’ ‘u’ 字母

③获取搜索框内容

④断言

预期返回值：XiaoLiu

实际返回值：XiaoLiu

Case7:

描述：测试全键盘数字在支付宝app登录界面的可用性

步骤：①打开支付宝app，点击手机号输入框

②依次点击‘1’ ‘2’ ‘3’ ‘4’ ‘5’ ‘6’ ‘7’ ‘8’ ‘9’ ‘0’ 数字

③获取输入框内容

④断言

预期返回值：123 4567 890

实际返回值：123 4567 890

Case8:（未通过）

描述：测试全键盘中文待选框词汇在百度app搜索框的可用性

步骤：①打开百度app，点击搜索框

②依次点击‘s’ ‘h’ ‘I’ ‘z’ ‘h’ ‘e’ ‘n’ ‘d’ ‘e’ ‘k’ ‘a’所对应的拼音

③点击待选框第一个词汇

④获取搜索框内容

⑤断言

预期返回值：是真的卡

实际返回值：是真的吗

Case9:（未通过）

描述：测试全键盘键入65个字母在微信app聊天框的可用性

步骤：①打开微信app，选择联系人，进入聊天框

②切换为英文模式，循环点击65次‘f’

③获取输入框内容，计算长度

④断言

预期返回值：65

实际返回值：60

Case10:

描述：测试全键盘中文输入在微信app聊天框的可用性

步骤：①打开微信app，选择联系人，进入聊天框

1. 随机生成两个字符的字符串，如：sh，点击sh，确认上屏
2. 点击shh,确认上屏
3. 点击shhh,确认上屏
4. 获取输入框内容
5. 点击发送
6. 断言

预期返回值：如：sh shh shhh

实际返回值：如：sh shh shhh

Class2：九宫格

Case1:

描述：测试九宫格键盘中文待选框词汇在哔哩哔哩app搜索框的可用性

步骤：①打开哔哩哔哩app，点击搜索框

②依次点击‘w’ ‘e’ ‘i’ ‘x’ ‘i’ ‘n’ 拼音

③获取搜索框内容

④断言

预期返回值：微信

实际返回值：微信

Case2:

描述：测试九宫格键盘英文字母在抖音app搜索框的可用性

步骤：①打开抖音app，点击搜索框

②切换为英文模式，依次点击键盘上26个字母

③获取搜索框内容

④断言

预期返回值：qwertyuiopasdfghjklzxcvbnm

实际返回值：qwertyuiopasdfghjklzxcvbnm

Case3:

描述：测试九宫格键盘英文输入长按操作在手机自带应用-便签的可用性

步骤：①打开便签，点击新建笔记

②切换为英文输入，依次长按 ‘z’ ‘x’ ‘c’ ‘v’ ‘b’ ‘n’ ‘m’ ‘@’ ‘,’

③获取文本内容

④断言

预期返回值："'~:;?!#."

实际返回值："'~:;?!#."

Case4:

描述：测试九宫格键盘英文大小写在网易云app搜索框的可用性

步骤：①打开网易云app，点击搜索框

②切换为英文模式，点击“大写”按钮，依次点击‘A’ ‘n’ ‘g’ ‘e’ ‘l’ ‘s’字母

③获取搜索框内容

④断言

预期返回值：Angels

实际返回值：Angels

Case5:

描述：测试九宫格键盘数字输入在QQ登录界面账号框的可用性

步骤：①打开QQ，点击账号框

②切换为数字模式，依次点击‘1‘ ’2‘ ’3‘ ’4‘ ’5‘ ’6‘ ’7‘ ’8‘ ’9‘ ’空格‘ ’@‘ ’0‘ ’.

③获取账号框内容

④断言

预期返回值：123456789 @0.

实际返回值：123456空格789@0.（但是手动测试时不会出现“空格”两字）

Case6:

描述：测试九宫格键盘标点符号在QQ登录界面密码框的可用性

步骤：①打开QQ，点击密码框

②切换为符号模式，依次点击常用符号

③获取密码框内容

预期返回值：获取点击对应的符号

实际返回值：获取点击对应的符号

Case7:

描述：测试九宫格键盘中文输入待选框第一个词汇在淘宝app搜索框的可用性

步骤：①打开淘宝app，点击搜索框

②依次点击‘l’ ‘i’ ‘a’ ‘n’ ‘y’ ‘i’ ‘q’ ‘u’ ‘n’ 拼音

③点击待选框第一个词汇

④获取搜索框内容

⑤断言

预期返回值：连衣裙

实际返回值：连衣裙

Case8:

描述：测试中文输入长按操作在微信app聊天框的可用性

步骤：①打开微信app，选择联系人，进入聊天框

②依次长按‘@#‘ ’ABC‘ ‘DEF’ ‘GHI’ ‘JKL’ ‘MNO’ PQRS’ ‘TUV’ ‘WXYZ’ 所对应的拼音

③获取输入框内容

④断言

预期返回值：1@#23456789

实际返回值：1@#23456789

Case9:（未通过）

描述：测试九宫格键盘键入65个字母在微信app聊天框的可用性

步骤：①打开微信app，选择联系人，进入聊天框

②切换为英文模式，循环点击65次‘f’

③获取输入框内容，计算长度

④断言

预期返回值：65

实际返回值：60

Case10:

描述：测试九宫格键盘中文待选框词汇在今日头条app搜索框的可用性

步骤：①打开今日头条app，点击搜索框

②依次点击‘w’ ‘e’ ‘i’ ‘x’ ‘i’ ‘n’ ‘j' ‘i’ ‘a’ ‘n’ ‘p’ ‘a’ ‘n’拼音

③获取搜索框内容

④断言

预期返回值：微信

实际返回值：微信

### 遇到的问题：

1. 有些app因为版本的更迭，某些元素的id值会发生变化，比如全民K歌的搜索按钮，在测试过程中，发生过三次改变，那如果使用id查询该元素的话，测试会报错，因此使用坐标查询，可以避免这种情况的出现。
2. Socket hang up

[Vivo 手机上运行 appium 自动化测试 出现 socket hang up 的问题解决 · TesterHome](https://testerhome.com/topics/22354)

# Appium自动化使用坐标定位元素

[葛木瓜](https://www.jianshu.com/u/ec370ab688d3)关注

2018.04.11 23:57:00字数 289阅读 6,104

### 1. 缘 由

为什么要用坐标定位元素呢？原因很简单，uiautomatorviewer无法定位该元素！

### 2. 场 景

目前我遇到无法定位的两种情况：

* 页面处于动态期间，如IM软件的视频或语音聊天期间，虽然元素可以正常定位，但是用click()无法触发点击事件
* 有些页面弹框上的元素，用uiautomatorviewer工具定位，只能定位某一片区域，具体的元素无法定位

### 3. 实 现

对于以上两种无法定位元素的情况，可以考虑用**坐标**定位

* 动态页面可以使用**swipe()**触发点击事件

# x1/y1和 x2/y2分别为起/止点的坐标，time为时间，单位为ms

driver.swipe(x1, y1, x2, y2, time)

'''

Usage：

driver.swipe(120, 200, 120, 200, 50)

定位坐标为(120, 200)的元素，并点击

'''

* 定位不到的元素可以使用**driver.tap()**触发点击事件

# 模拟手指点击（最多五个手指），time为时间，单位为ms

driver.tap([(x1, y1), (x2, y2)], time)

'''

Usage：

driver.tap([120, 200], 10)

定位坐标为(120, 200)的元素，并点击

'''

### 4. 问 题

以上两种坐标定位都使用元素的**绝对坐标**，如更换模拟器分辨率，元素坐标也会随之**改变**，此时就需要将当前分辨率下的**绝对坐标转换为相对坐标**，方法如下：

# 获取当前分辨率

el\_x = driver.get\_window\_size()['width']

el\_y = driver.get\_window\_size()['height']

# 绝对坐标转换为相对坐标，假设当前分辨率为720x1280，绝对坐标为(120, 200)

xd\_x = (120/720)\*el\_x

xd\_y = (200/1280)\*el\_y

>> (xd\_x, xd\_y) # 相对坐标

### 5.参 考

* [appium 常用api介绍（2）](https://link.jianshu.com/?t=http%3A%2F%2Fwww.cnblogs.com%2Folivexiao%2Fp%2F6134823.html)
* [appium使用相对坐标定位元素](https://link.jianshu.com/?t=https%3A%2F%2Fblog.csdn.net%2Fqq_37695583%2Farticle%2Fdetails%2F79320116)

2，通过相对坐标定位（推荐）

更换不同分辨的测试机后，之前获取的手机某个位置点的绝对坐标就会改变，这样每次更换测试机都需要更改坐标，显然是不方便代码的维护，所以就需要获取相对坐标。

2.1，获取相对坐标的思路：

1，选择任意一台测试手机（或模拟器）

2，获取需要定位的点的绝对坐标(x1, y1)

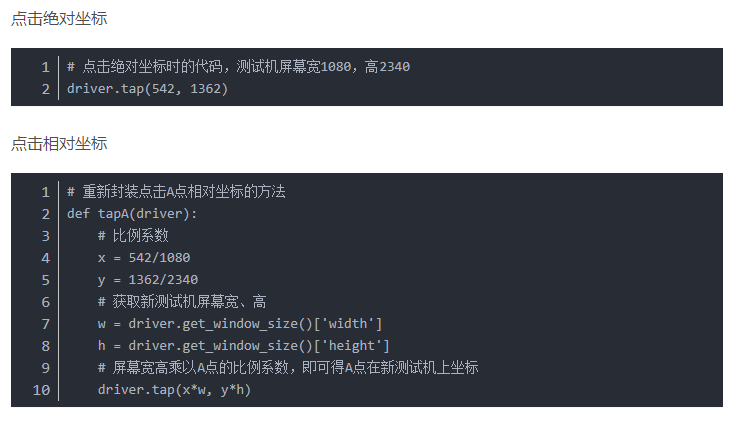
3，获取当前测试机屏幕的宽度(w1)、高度(h1)

4，获取该点横坐标与屏幕宽的比(x1/w1)，纵坐标与屏幕高的比(y1/h1)

5，更换测试机后，获取新的测试机屏幕的宽(w2)，高(h2)

6，计算新测试机上该点横坐标x2=w2 \* (x1/w1)，纵坐标y2 = h2 \* (y1/h1)

2.2，点击A点代码如下：



需要注意：

1，webdriver中提供坐标点击方法tap()，而元素点击方法则是click()；

2，区别于TouchAction类中的tap()方法，不需要构造TouchAction对象，driver对象调用即可。

————————————————

版权声明：本文为CSDN博主「蜀山客e」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

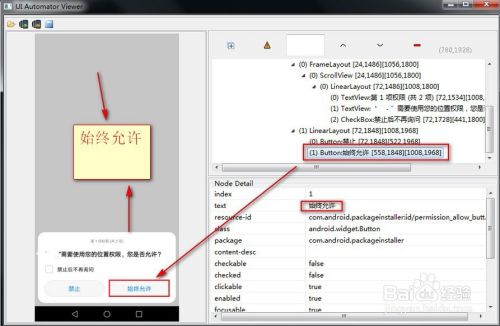
原文链接：<https://blog.csdn.net/Asaasa1/article/details/109401154>

# 弹窗处理

1. 在安装app首次启动的时候，在引导页出现权限弹出框



2.弹窗是可以直接定位到的，这里不能通过id去定位了，因为弹窗比较多，每次id都不太一样，但是文本text都是一样的：始终允许



1. 通过xpath来定位按钮：始终允许代码