

类自然语言 Android自动化测试

目录

- ✿ 我们的故事
- ✿ 举个例子
- ✿ 特点和经验

我们的故事

录制 + 图像对比 + C#

- 速度慢、准确性差
- 入门者只会录制，进阶到熟练C#难度很大

Espresso + Java

- 入了自动化的门
- 愿学

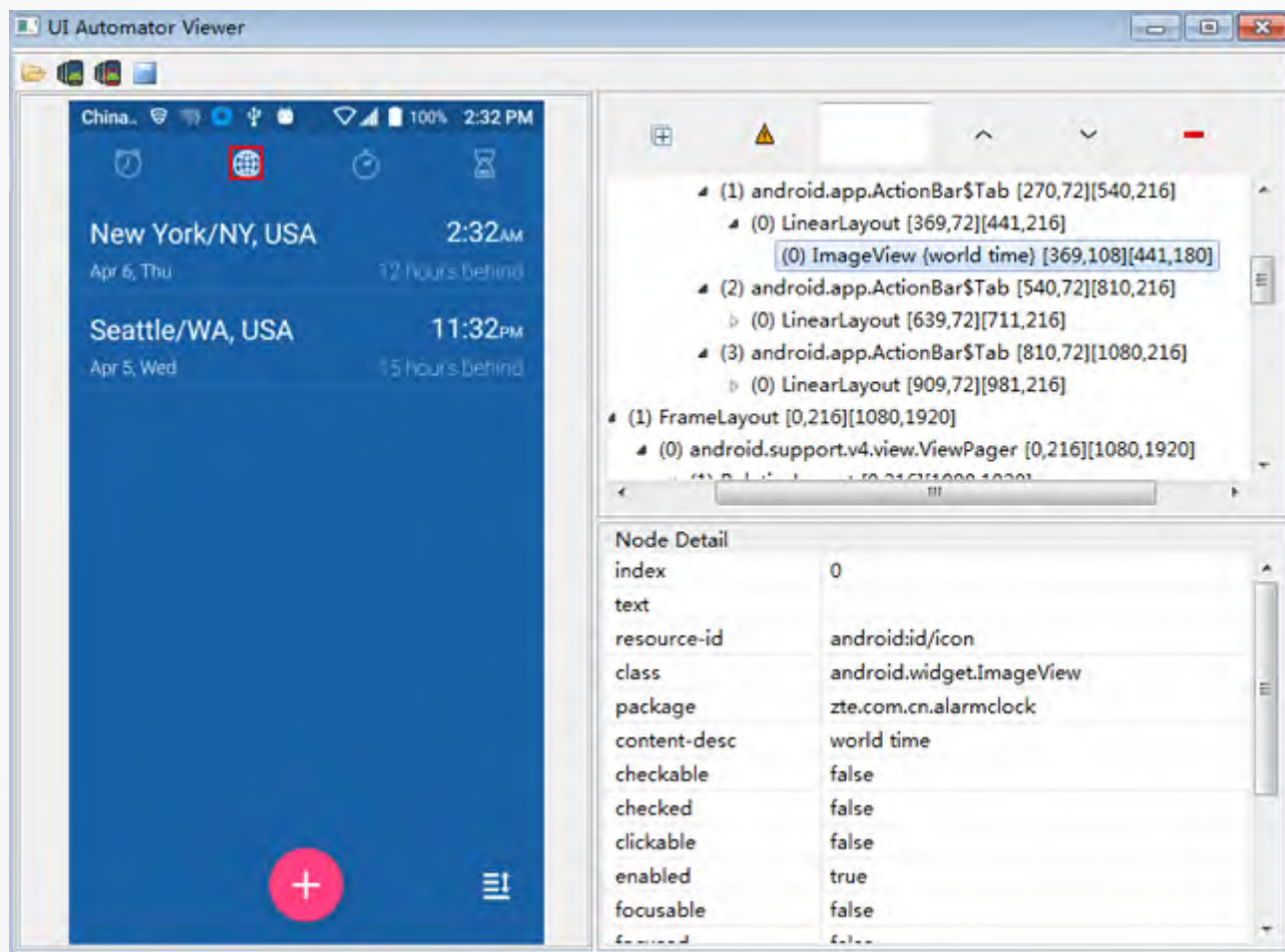
UiAutomator + 类自然语言

- 入门易，吸引了一批新人
- 不从录制开始
- 了解基础，逐步提升
- python + UiAutomator

我们的故事

- 封装了一个自己的简易框架，用自然语言和类自然语言写脚本，大大降低了入门难度
- 考虑实际的使用场景：整机测试、定制多。针对性考虑脚本的通用性
- 支持通过python进行扩展，使不同人各尽其能

用例基础



对照UI Automator Viewer无困难编写/阅读用例

Show me the code

```
1 //1. 定义用例:
2 case 增加世界时显示
3     打开时钟应用
4     选择世界时界面
5     选择添加世界时
6     选择增加西雅图时间
7     确认世界时增加成功
8 endcase
9
10 //2. 定义用到的功能块、状态和动作等:
11 打开时钟应用, shell am start -n zte.com.cn.alarmclock/.AlarmClock
12 选择世界时界面, click 世界时界面标签
13 世界时界面标签, ui:content-desc=world time
14 选择添加世界时, c@id:globaltime_add_button
15 选择增加西雅图时间, c@uv:text=Seattle
16
17 sub 确认世界时增加成功
18     当前在世界时界面
19     西雅图显示在世界时列表中
20 endsub
21
22 当前在世界时界面, wait 世界时界面标签
23 西雅图显示在世界时列表中, w@ui:text=Seattle
```


用例运行

```

2017-04-06 14:23:02,362 - I - Trying to lock phone: d3bbda8e
2017-04-06 14:23:02,659 - I - 增加世界时显示: START
2017-04-06 14:23:02,864 - I - install: d:\Work\UiMap\UiMapNew\uimapnew\uimap\libs\uimap.apk
2017-04-06 14:23:04,537 - I - install: d:\Work\UiMap\UiMapNew\uimapnew\uimap\libs\uimap-test.apk
2017-04-06 14:23:15,347 - I - I/UiMap (17884): setup
2017-04-06 14:23:15,377 - I - I/UiMap (17884): -----
2017-04-06 14:23:15,378 - I - I/UiMap (17884): STEP: 打开时钟应用
2017-04-06 14:23:15,378 - I - I/UiMap (17884): =>ACTION: shell:am start -n zte.com.cn.alarmclock/.AlarmClock
2017-04-06 14:23:17,497 - I - I/UiMap (17884): success
2017-04-06 14:23:17,498 - I - I/UiMap (17884): -----
2017-04-06 14:23:17,498 - I - I/UiMap (17884): STEP: 选择世界时界面
2017-04-06 14:23:17,498 - I - I/UiMap (17884): =>ACTION: click@ui:content-desc=world time
2017-04-06 14:23:19,264 - I - I/UiMap (17884): success
2017-04-06 14:23:19,265 - I - I/UiMap (17884): -----
2017-04-06 14:23:19,265 - I - I/UiMap (17884): STEP: 选择添加世界时
2017-04-06 14:23:19,266 - I - I/UiMap (17884): =>ACTION: click@ui:id=globaltime_add_button
2017-04-06 14:23:21,082 - I - I/UiMap (17884): success
2017-04-06 14:23:21,083 - I - I/UiMap (17884): -----
2017-04-06 14:23:21,083 - I - I/UiMap (17884): STEP: 选择增加西雅图时间
2017-04-06 14:23:21,133 - I - I/UiMap (17884): =>ACTION: click@uv:text=Seattle
2017-04-06 14:23:31,403 - I - I/UiMap (17884): success
2017-04-06 14:23:31,403 - I - I/UiMap (17884): -----
2017-04-06 14:23:31,404 - I - I/UiMap (17884): STEP: 确认世界时增加成功
2017-04-06 14:23:31,404 - I - I/UiMap (17884): =>ACTION: wait@ui:content-desc=world time
2017-04-06 14:23:33,034 - I - I/UiMap (17884): success
2017-04-06 14:23:33,034 - I - I/UiMap (17884): =>ACTION: wait@ui:text=Seattle
2017-04-06 14:23:33,922 - I - I/UiMap (17884): success
2017-04-06 14:23:33,926 - I - I/UiMap (17884): setup
2017-04-06 14:23:34,224 - I - 增加世界时显示: PASS
2017-04-06 14:23:34,441 - I - Release phone: d3bbda8e
2017-04-06 14:23:34,742 - I - finished successfully
  
```

特点：对比成熟BDD框架

- 同为自然语言
- 不需要先掌握一门编程语言
- 能用类自然语言完成绝大多数的工作
- 劣势：规范较松，没有用严格的
given/when/then

特点：对比录制优先的工具

- 需要明白ui布局、控件标识这些适当的细节
- 需要写最简单的控制语句
- 有助于脚本调试和逐步建立自动化脚本编写的概念
- 实践中，很多同事通过这些初步的脚本编写转而学习python，具备了更多的编程能力

特点：语法迭代

- 脚本关键字、脚本描写形式根据实际使用需求不断迭代
- `onView(withText("World")).perform(click());`
- 易于阅读，但前提是英文够好
- 我们提供了这样的：`click ui:text=World`
- 最后大家最喜欢这样的：`c@tt:World`

特点 - 项目兼容处理

- 小差异：或操作

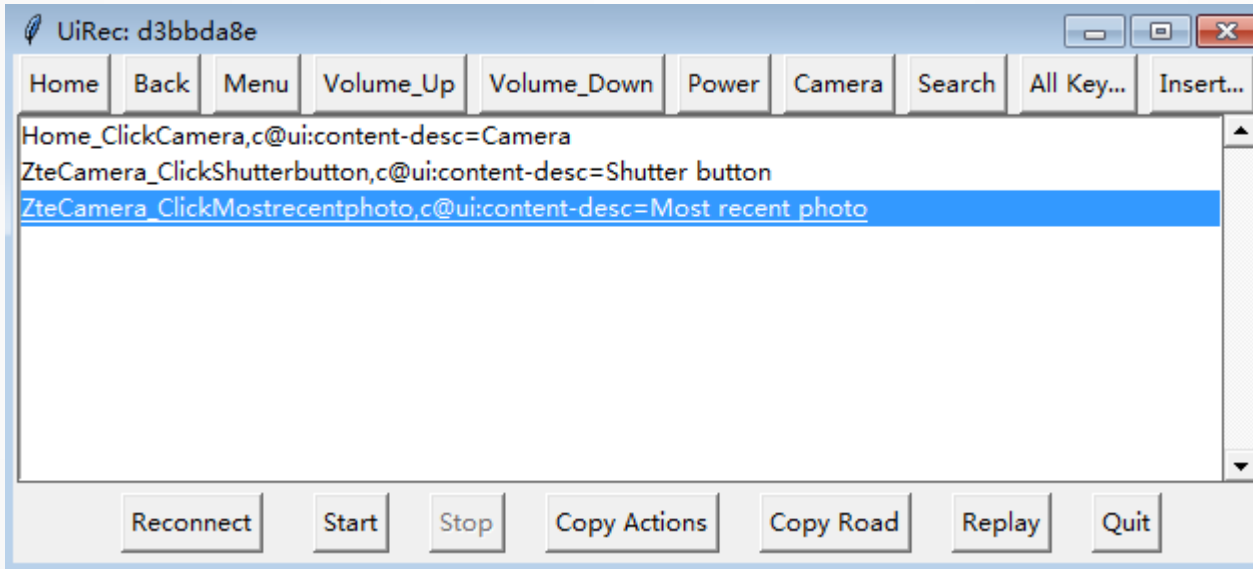
```
保存闹钟,c@id:tv_done||c@id:alarm_ok||c@ui:text=save
```

- 中型差异：if 操作

```
...  
if 是联通卡  
    拨打电话给联通客服  
else  
    拨打电话给电信客服  
endif  
...
```

- 大差异：用例filter

特点 - 简单的录制回放



- 不同于dump/截图式的录制方法。这里的录制基于真机操作时的events解析，实时无延迟
- 部分操作未做支持，仅作为脚本编写的辅助手段

特点 – python脚本接口

- 通过与python函数的无缝结合，扩展框架的能力
 - 定义一个python函数，用于设置AP的加密方式：

```
def SetAPtoWPA_PSK_TKIP_MODE(main, phone, log, var):  
    ...
```

- 在用例中直接使用该函数，和使用其他动作一样：

```
Route 连接WPA_PSK_TKIP方式热点  
SetAPtoWPA_PSK_TKIP_MODE  
设置界面连接热点  
...
```

特点 - 与其他框架结合

- 作为一个功能库和Behave, Robot Framework等框架结合使用
- 以python behave为例:
 - 构建feature文件
 - 配合一个通用的step文件

```
Feature: 世界时

Scenario: 增加世界时显示
Given 手机处于桌面
  When 打开时钟应用
  And 选择世界时界面
  And 选择添加世界时
  And 选择增加西雅图时间
  Then 世界时增加成功
```

```
import sys
import os
from behave import *
from uimap.__main__ import *

folder = os.path.dirname(__file__)
LoadMap(folder, recursively=True)

@given('{text}')
def step_impl(context, text):
    assert Walk(text)

@when('{text}')
def step_impl(context, text):
    assert Walk(text)

@then('{text}')
def step_impl(context, text):
    assert Walk(text)
```


特点 - 与其他框架结合

- 运行

```
$ behave
```

```
Feature: 世界时
```

```
Scenario: 增加世界时显示
```

```
Given 手机处于桌面
```

```
2017-04-06 20:17:59,125 - I - I/UiMap (17210): setup
2017-04-06 20:17:59,165 - I - I/UiMap (17210): =>ACTION: wait@ui:package=launcher
2017-04-06 20:18:00,536 - I - I/UiMap (17210): success
```

```
Then 世界时增加成功
```

```
3 KB/s (113 bytes in 0.031s)
```

```
2017-04-06 20:18:27,044 - I - I/UiMap (17517): setup
2017-04-06 20:18:27,078 - I - I/UiMap (17517): =>ACTION: wait@ui:content-desc=world time
2017-04-06 20:18:28,325 - I - I/UiMap (17517): success
2017-04-06 20:18:28,327 - I - I/UiMap (17517): =>ACTION: wait@ui:text=Seattle
2017-04-06 20:18:29,029 - I - I/UiMap (17517): success
2017-04-06 20:18:29,032 - I - I/UiMap (17517): setup
2017-04-06 20:18:29,276 - I - success
```

```
1 feature passed, 0 failed, 0 skipped
1 scenario passed, 0 failed, 0 skipped
6 steps passed, 0 failed, 0 skipped, 0 undefined
Took 0m46.912s
```

特点 - 用例自适应

- 用例Tag: 自定义用例标签, 如等级(p1, p2...)、模块(Camera, Music)等
- 用例Filter: 类对JUnit的Assume机制的一种实现
 - 对每条用例按其真实的适配情况来进行配置, 如有些用例适合某个Android平台, 有些用例适合某些运营商, 有些用例只适合某个项目
 - 执行时通过对测试机的属性查询(getprop), 让每条用例自己决定是否是否需要运行

特点 - 用例自适应

- 例:

```
route ABC
'''
@Type:Function
@Module:Camera
@allow ro.product.name:abc
@allow ro.build.version.release:6
@deny persist.carrier.activated_id:SPRINT
'''
A
```

- 在不同项目运行时可以用**相同命令**启动（如：--tag smoke 运行所有smoke用例），用例的选择会动态调整，在不同项目上会执行不同的用例

```
Route 将audio文件设置为通知铃声 is skipped(ro.build.version.sdk: 23)
Route GIF制作 is skipped(no filter matched)
```

实现方式简介

PC侧

- Python脚本实现
- 负责测试启动、文件解析、用例管理
- 负责python自定义功能、PC侧命令的执行，将手机侧命令按标准格式传递到手机端
- 负责log获取，测试报告输出

手机侧

- UiAutomator编写，负责解析手机侧的动作并执行
- 支持Loop/If操作，可以减少与PC的交互，缩短用例运行时间