UiAutomator UiWatcher类API（七）

By Zhang Jingwei

20160311

概要

- UiWatcher类介绍
- 中断监听检查条件
- 监听器操作
- 检查监听器

UiWatcher类介绍

1.UiWatcher类说明

- UiWatcher是用于处理用例执行过程中遇到非预想的情况

2.UiWatcher使用场景

- 测试过程中来了一个电话
- 测试过程中来了一条短信
- 获取测试过程中来了一个视频通话
- 出现各种非预想情况

中断监听检查条件

1.public boolean checkForCondition()

- 注册UiWatcher必须实现的一个方法

2.用处

- uiautomator框架通过UiSelector函数无法获取到UI控件的时候，就会调用在UiDevice上注册的UiWatcher的checkForCondition函数。

示例

```
UiDevice.getInstance().registerWatcher("demo", new UiWatcher(){
@Override
public boolean checkForCondition() {
// TODO Auto-generated method stub
return false;
}
});
```

监听器操作

1. 相关API

返回值	方法名	描述
void	registerWatcher(String name, UiWatcher watcher)	注册一个监听器，当UiSelector无法找到对象时触发监听器
void	removeWatcher(String name)	取消之前注册过的监听器
void	resetWatcherTriggers()	重置已触发过的监听器，重置之后相当于没运行过
void	runWatchers()	强制运行所有的监听器

检查监听器

1. 相关API

返回值	方法名	描述
void	hasAnyWatcherTriggered()	检查是否有监听器触发过
void	hasWatcherTriggered(String watcherName)	检查某个特定触发器是否触发过

示例

```
public void mRegisterWatcher(String name){
UiDevice.getInstance().registerWatcher(name, new UiWatcher(){
//挂断微信视频聊天邀请
UiObject hangUpBtn=new UiObject(new UiSelector().text("Hang Up"));
@Override
public boolean checkForCondition() {
System.out.println("监听器开始运行");
if(hangUpBtn.exists()){
System.out.println("监听器条件判断成功");
UiDevice.getInstance().click(226,1534);
return true;
}
System.out.println("监听器条件判断失败");
return false;
}
});
}
```

```
public void testWatcherActionsAndCheck() throws UiObjectNotFoundException{
//UiWatcher, Wechat video call
mRegisterWatcher("VideoCall");//RegisterWatcher
//Wechat send message
UiObject inputMsg=new UiObject(new UiSelector().resourceIdMatches(".*id/wh"));
UiObject sendMsg=new UiObject(new UiSelector().resourceIdMatches(".*id/wn"));
inputMsg.setText("1");
sleep(2000);
sendMsg.click();
sleep(2000);
inputMsg.setText("2");
sleep(2000);
sendMsg.click();
sleep(2000);
inputMsg.setText("3");
sleep(2000);
sendMsg.click();
sleep(2000);
inputMsg.setText("4");
sleep(2000);
sendMsg.click();
sleep(2000);
```

```
//hasAnyWatcherTriggered
boolean hasAnyWatcherTriggered=UiDevice.getInstance().hasAnyWatcherTriggered();
System.out.println("hasAnyWatcherTriggered:::"+hasAnyWatcherTriggered);
//hasWatcherTriggeredVC
boolean hasWatcherTriggeredVC=UiDevice.getInstance().hasWatcherTriggered("VideoCall");
System.out.println("hasWatcherTriggeredVC:::"+hasWatcherTriggeredVC);
sleep(500);
UiDevice.getInstance().removeWatcher("VideoCall");//removeWatcher
inputMsg.setText("5");
sleep(2000);
sendMsg.click();
sleep(2000);
inputMsg.setText("6");
sleep(2000);
sendMsg.click();
sleep(2000);
inputMsg.setText("7");
sleep(2000);
sendMsg.click();
sleep(2000);
```

```
inputMsg.setText("8");
sleep(2000);
sendMsg.click();
sleep(2000);
}
```

Q&A