Uiautomator资料整理

uiautomator介绍

> The uiautomator testing framework lets you test your user interface (UI) efficiently by creating automated functional UI testcases that can be run against your app on one or more devices. ————Google

---

语法

> adb shell uiautomator runtest ()jar) -c (test_class_or_method)_ [options]

例：

adb shell uiautomator runtest LaunchSettings.jar -c com.uia.example.my.LaunchSettings

说明：
adb shell uiautomator runtest ————————既定格式
********** .jar ————————————JAVA生成的包

> Required. The argument is the name of one or more JAR files that you deployed to the target device which contain your uiautomator testcases. You can list more than one JAR file by using a space as a separator.

-c (test_class_or_method) ——JAR中的类与方法

> Required. The argument is a list of one or more specific test classes or test methods from the JARs that you want uiautomator to run.
> Each class or method must be fully qualified with the package name, in one of these formats:
>
> package_name.class_name
> package_name.class_name#method_name
> You can list multiple classes or methods by using a space as a separator.

–nohup

> Runs the test to completion on the device even if its parent process is terminated (for example, if the device is disconnected).

-e name value

> Specify other name-value pairs to be passed to test classes. May be repeated.
> 例如：adb shell uiautomator runtest test.jar -name “Tom” -c **

就是往其中传递参数的，这些参数是可配置的,而且参数必须是字符串类型。

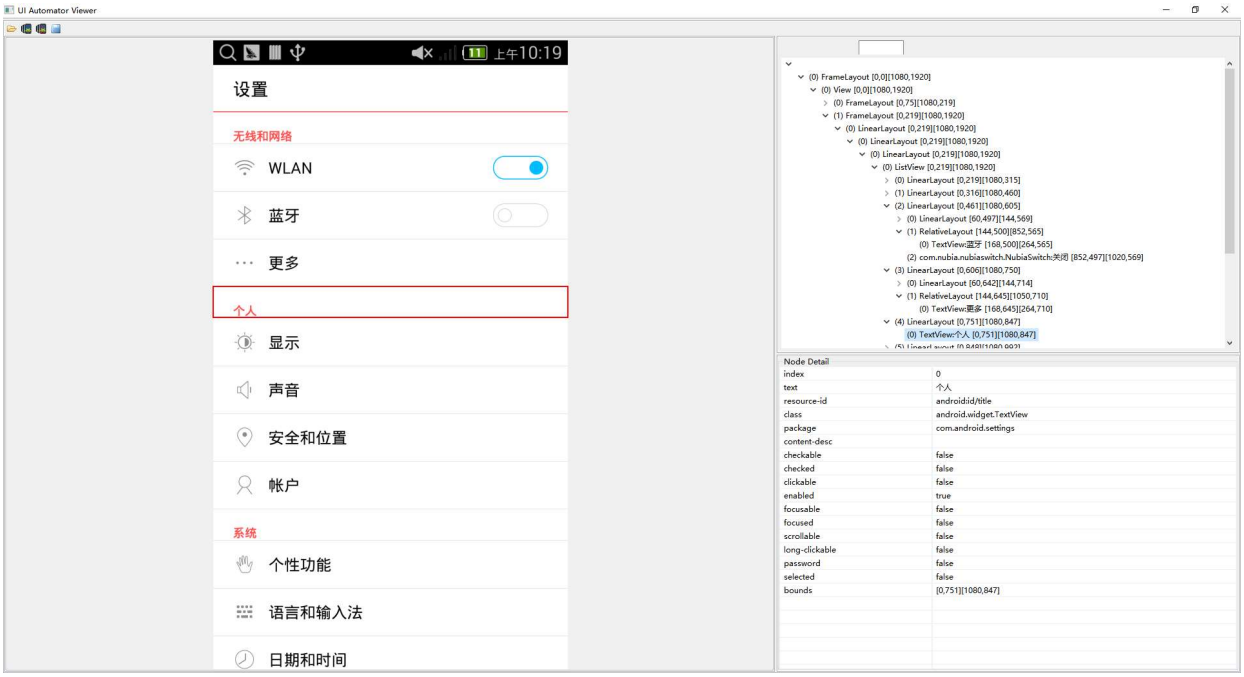
在JAVA中的调用方法：

```
1. private static String name;
2. //先定义好其中的参数
3.
4. Bundle bundle=getParams();
5. name=bundle.getString("appname");
6.
7. //appname就是在使用命令行时的名称：
8. // - e  appname  "Tome"
9.
```

Uiautomator 工具

> The uiautomator API is bundled in the uiautomator.jar file under the /platforms/ directory. The API includes these key classes, interfaces, and exceptions that allow you to capture and manipulate UI components on the target app:

Uiautomatorviewer的所在路径：
your_sdk\tools\uiautomatorviewer.bat
双击就可以使用，但是不可以关闭
在这个目录下还有许多的工具，但是我们现在只需要这一种就可以了
Uiautomator可以显示电脑连接的Android设备的截图，里面清晰的展示了APP的布局



1.左边时应用截图
2.右上角时布局的方式
3.右下角是元素的属性：

> index:索引号
> text:文本
> resource-id:资源ID
> class:类名
> content-desc:内容描述

Uiautomator API

Classes:

1.com.android.uiautomator.core.UiCollection
提供了设备UI的元素，子元素以及它们的描述
2.com.android.uiautomator.core.UiDevice
提供了设备的信息，可以使用此类来模拟一些操作，比如锁屏，按Home键等操作
3.com.android.uiautomator.core.UiObject
代表一个UI元素
4.com.android.uiautomator.core.UiScrollable
提供一个搜索元件和滚动容器的支持
5.com.android.uiautomator.core.UiSelector
在设备截图上寻找一个或多个UI元素

Interfaces

1.com.android.uiautomator.core.UiWatcher
在目标设备上提供了有条件的监视器
2.com.android.uiautomator.testrunner.IAutomationSupport
为运行测试用例提供了辅助支持
3.com.android.uiautomator.testrunner.UiAutomatorTestCase
为运行多个测试用例限定了环境。所有的uiautomator测试用例都应该继承这个类

Exceptions

1.com.android.uiautomator.core.UiObjectNotFoundExcception
表示当一个UiSelector寻找不到指定的UI元素

详细说明

IAutomationSupport
//输出到运行报告中：
Public Methods:
sengStatus(int resultCode,Bundle status)

UiAutomatorTestCase
概述：
自动化测试都继承自这个类，主要提供以下两个方法：
1.Uidevice instance 设备实例
2.Bundle for command line parameters 捆绑命令行参数
public Methods:
1.IAutomationSupport —getAutomationSupport()

> public IAutomationSupport getAutomationSupport ()

2.Bundle ——getParams()

> public Bundle getParams()

```
1. Bundle bundle =getParams();
2. appname=bundle.getString("appname");
```

3.UiDevice ——getUidevice()

> public UiDevice getUiDevice()
> //下面解释

4.sleep(long ms)

> public void sleep (long ms)

```
1. sleep(1000);  //休眠1秒
```

UiCollection

概述：
提供了设备UI的元素，子元素以及它们的描述
Public Methods：
1.getChildByDescription(UiSelector childPattern,String text)
在UiSelector的先定中获得元素的内容描述
2getChildByInstance(UiSelector childPattern,int insance)
在UiSelector的先定中获得元素在集合中位置

> 说明：Instance,这是一个很重要方法，如果有多个元素的class,index ,text属性是一样的，而instance则是把它们放在一个集合中（以0开始），用来查找。

3.getChildByText(UiSelector childPattern ,String text)
在UiSelector的先定中获得元素的内容

4.getChildCount(UiSelctor childPattern) —-return int
得到UiSelector中的元素的个数,只能够是当前可视的控件数目，并且滚动条是关闭的。

```
1. UiObject uiobject=new UiObject(new UiSeclector().className("android.wid
   get.ListView"));
2.
3. int i =uiobject.getChildCount(new UiSelector().className("android.widge
   t.LinearLayout"))
4.
5. //获得当前的列表中可见的元素个数总和，并不一定是这个列表全部的，知识当前的屏幕可见
   的。
```

UiDevice

概述：
提供了设备的信息，可以使用此类来模拟一些操作，比如锁屏，按Home键等操作
目录：返回类型————方法
void————————————clearLastTraversedText()
从最后一次的UI遍历事件中清除文字
boolean————————————click(x,y)
点击一个任意指定的坐标
void————————————dumpWindowHierarchy(String fileNmae)
帮助方法去调试当前的窗口布局方式
void————————————freezeRotation
禁用传感器并冻结设备当前旋转状态下的旋转。
String————————————getCurrentPackageName()
得到当前的包名
int————————————getDisplayHeight()
获取显示的高度（以像素为单位）
int————————————getDisplayWidth()
获取显示的宽度（以像素为单位）
static UiDevice ————getInstance()
检索一个设备的单例

```
1. UiDevice.getInstance.click();
2. //在使用UiDevices时，最好都要带上这个，否则有时会报错
```

String ————————getLastTraversedText()
从最后一次的UI遍历事件记录中检索文本
String————————getProductName()
检索这个设备产品的名称
boolean————————hasAnyWatcherTriggered()
检查是否有任何监视被触发
boolean————————hasAnyWatcherTriggered(String fileName)
检查一个明确的监视器是否被触发
boolean————————isNaturalOrientation()
检查设备是否处于原来的方向
boolean————————isScreenOn()
如果屏幕关闭，检查电源键是否开启
boolean————————pressBack()
模拟一个短按返回按键
boolean————————pressDPadCenter()
模拟一个短按中间按键
boolean————————pressDpadDown()
模拟一个短按向下按键
boolean————————pressDPadLeft()
模拟一个短按左键
boolean————————pressDPadRight()
模拟一个短按右键
boolean————————pressDPadUp()
模拟一个短按向上按键
boolean————————pressDelete()
模拟一个短按删除键
boolean————————pressEnter()
模拟一个短按的确认键
boolean————————pressHome();
模拟一个短按Home键
boolean————————pressMenu();
模拟一个短按菜单键
boolean————————pressRecentApps()
模拟一个短按显示最近APP的按键
boolean————————pressSearch()
模拟一个短按搜索键
boolean————————pressKeyCode(int keyCode)
模拟一个短按键码
boolean————————pressKeyCode(int keyCode,int metaState)
模拟一个短按键码
void ————————registerWatcher(String name,UiWatcher watcher)
注册监视器
void————————removeWatcher(String name)
移除监视器
void————————runWatchers()
强制所有的监视器运行
void————————setOrientationLeft()

模拟将设备定向到左侧，并通过禁用传感器来冻结旋转
void————————————setOrientationNatural()
模拟将设备定向为其自然方向，并通过禁用传感器来冻结旋转
void ————————————setOrientationRight()
模拟将设备定向到右侧，并通过禁用传感器来冻结旋转
void————————————sleep()
如果屏幕是亮着的，这个方法模拟按电源键，如果屏幕是关着的则没有用处
boolean————————————swipe(Point[]segments,int segmentSteps)
在点阵之间移动，默认步幅为5
boolean————————————swip(int startX,int startY,int endX,endY,int steps)
从一个坐标滑动到另一个坐标，默认步幅为5
boolean————————————takeScreenshot(File strorePath,float scale,int quality)
截图：

> storePath: where the PNG should be written to
> scale: scale the screenshot down if needed; 1.0f for original size
> quality: quality of the PNG compression; range: 0-100

boolean————————————takeScreenshot(File storePath)
截图：PNG格式，默认大小为1，质量为百分之90
void————————————unfreezeRotation()
重新启用传感器并取消冻结设备旋转，允许其内容随设备物理旋转而旋转。 在测试执行期间，最好保持设备冻结在特定方向，直到测试用例执行完成。
void————————————waitForIdle(long time)
在一段时间内等待当前的应用空闲
void————————————waitForWindowsUpdate(String package,long time)
等待当前的窗口内容发生更新，如果这个包名是明确的，但是当前的窗口并没有相同的包名，则这个函数立即返回
void————————————wakeUp()
唤醒设备

UiObject

概述：

> A UiObject is a representation of a user interface (UI) element. It is not in any way directly bound to a UI element as an object reference. A UiObject holds information to help it locate a matching UI element at runtime based on the UiSelector properties specified in its constructor. Since a UiObject is a representative for a UI element, it can be reused for different views with matching UI elements.

目录：返回类型———方法
void ————————clearTextField()
在编辑框中清除已经存在的内容
boolean————————click()
在可视元素的中心点击
boolean————————clickAndWaitForNewWindow(long timeout)
点击并且等待窗口转换，timeout为超时时间
boolean————————clickAndWaitForNewWindow()
点击等待窗口更新
boolean————————clickBottomRight()
点击元素的右下角
boolean————————-clickTopLeft()
点击元素的右上角
boolean————————-exists()
检查元素是否存在
Rect————————————getBounds()
返回ui元素的明确边界
UiObject————————getChild(UiSelector slector)
创建一个新的UiObject来代表当前的子元素
int————————————getChildCount()
计算当前的uiobject所代表的元素的子元素的数量
String————————getContentDescription()
获得元素的内容描述
Uiobejct————————getFromParent(UiSelector selector)
从当前的父元素中创建一个新的UiObject来代表这个子Ui元素
String————————getPackageName()
读取这个元素的包属性
String————————getText()
获得元素的文本
Rect————————————getVisibleBounds()
获取UI元素的可见边界
boolean————————isCheckable()
检查当前的元素是否时可检查的
boolean————————isClickable()
检查当前的属性是否时可点击的
boolean————————isEnable()
检查元素的enable属性是否时正确
boolean————————isScrollable()
检查元素的滚动属性
boolean————————setText(String text)
输入文本
boolean————————swipeDown(int steps)
向下滑动
boolean————————swipeRight(int steps)
向左滑动
boolean————————swipeLeft(int steps)
向右滑动

boolean————————swipeUp(int steps)
向上滑动
boolean————————waitForExists(long timeout)
在一个明确的时间段内等待一个Ui元素可见
boolean————————waitUntilGone(long timeout)
在一个明确的时间内等待一个UI元素无法检测到\

UiScrollable

概述：

> UiScrollable is a UiCollection and provides support for searching for items in a scrollable user interface (UI) elements. This class can be used with horizontally or vertically scrollable controls..

构造函数：
UiScrollable(UiSelector container)

方法：
boolean————————flingBackward()
执行一个向后滑动的操作，如果时垂直的方向，则从顶部滑动到底部，如果时水平的方向，则从左边滑动到右边。
boolean————————flingForWard()
一个简易版的scrollForward(int),向前滑动的操作
boolean————————flingToBeginning(int maxSwipes)
以最大的步幅滑动到顶部
boolean————————flingToEnd(int maxSwipes)
以最大的步幅滑动到底部
UiObject————————getChildByDescription(UiSelector childPattern, String text, boolean allowScrollSearch)
在此UiScrollable UiSelector容器的约束内搜索子UI元素
UiObject————————getChildByInstance(UiSelector childPattern, int instance)
在此UiScrollable UiSelector容器的约束内搜索子UI元素
UiObject————————getChildByText(UiSelector childPattern, String text, boolean allowScrollSearch)
在此UiScrollable UiSelector容器的约束内搜索子UI元素
UiObject————————getChildByText(UiSelector childPattern, String text)
在此UiScrollable UiSelector容器的约束内搜索子UI元素
int————————getMaxSearchSwipes()
获得最大的搜索步幅
double————————getSwipeDeadZonePercentage()
当滑动时获得小部件未点击区域的百分比
boolean————————scrollBackward(int steps)
执行一个向后滑动的操作，默认步幅未5，括号内可不太填
boolean————————scrollDescriptionIntoView(String text)
在UI元素上执行向上滑动，直到请求的内容描述可见或直到滑动尝试耗尽
boolean————————scrollForward()
执行一个向前滑动的操作，默认步幅未5，括号内可不太填
boolean————————scrollIntoView(UiSelector selector)
执行一个匹配UiSelector的滑动寻找操作
boolean————————scrollTextIntoView(String text)
在UI元素上执行向上滑动，直到请求的文本可见或直到滑动尝试耗尽
boolean————————scrollToBeginning(int maxSwipes)
滑动到可滑动元素的顶部
boolean————————scrollToBeginning(int maxSwipes, int steps)
滑动到可滑动元素的顶部
boolean————————scrollToEnd(int maxSwipes, int steps)
滑动到可滑动元素的底部
boolean————————scrollToEnd(int maxSwipes)
滑动到可滑动元素的底部
void————————setAsHorizontalList()
设置滚动方向为水平方向
void————————setAsVerticalList()
设置滚动方向为垂直方向
void ————————setSwipeDeadZonePercentage(double

swipeDeadZonePercentage)
设置小部件的大小在滑动时被视为无接触区域的百分比。

UiSelector

概述：

> This class provides the mechanism for tests to describe the UI elements they intend to target. A UI element has many properties associated with it such as text value, content-description, class name and multiple state information like selected, enabled, checked etc. Additionally UiSelector allows targeting of UI elements within a specific display hierarchies to distinguish similar elements based in the hierarchies they're in
> 这个类提供了测试的机制来描述他们打算定位的UI元素。 UI元素具有与其相关联的许多属性，例如文本值，内容描述，类名称以及诸如选择，启用，检查等的多个状态信息。另外，UiSelector允许在特定显示层级中定向UI元素以基于它们所处的层次结构

构造函数：
UiSelector()

方法：
UiSelector————————childSelector(UiSelector selector)
向此选择器添加子UiSelector条件
UiSelector————————className(String className)
设置以部件类的属性作为搜索的标准
UiSelector————————classNameMatches(String regex)
设置以部件类的属性作为搜索的标准(正则表达式)
UiSelector————————clickable(boolean val)
设置以部件是否能点击的属性作为搜索的标准
UiSelector————————description(String desc)
描述
UiSelector————————descriptionContains(String desc)
描述包含
UiSelector————————descriptionMatches(String regex)
正则表达式内容描述
UiSelector————————descriptionStartsWith(String desc)
开头内容描述匹配
UiSelector————————text(String text)
文本
UiSelector————————textContains(String text)
文本包含
UiSelector————————textMatches(String regex)
文本正则表达式
UiSelector————————textStartsWith(String text)
在显示器上可见的文本

UiWatcher

概述:
监视器，如果在测试的过程中出现了意外的控件，比如升级界面，并且时随机出现的，那么就可以使用监视器随时监视异常控件的出现，并且做出一定的操作。

Uiautomator的限制

1.无法对webview中的元素进行定位
2.需要android版本号在4.3或更高的版本（API18—>）

如何使用

1.android create uitest-project -n demo.jar -t 1 -p ###
#表示java-projeect所在的路径
2.ant build
必须进入到java-projeact中使用命令
3.进入到bin目录中
adb push demo.jar /data/local/tmp
4.adb shell uiautomator runtest demo.jar -c ^^^#^^^
^^^#^^^^必须准确到哪一个方法.(用来测试单个的方法)