UiAutomator UiDevice类API（二）

By Zhang Jingwei

20160202

概要

- UiDevice类介绍
- 按键与Keycode使用
- 获取坐标与坐标点击
- 拖拽与滑动
- 旋转屏幕
- 灭屏与唤醒屏幕
- 截图与等待空闲
- 获取包名&开启通知栏&快速设置&获取布局文件

UiDevice类介绍

- UiDevice代表设备状态
- UiDevice为单例模式
- UiDevice功能

  1）获取设备信息：屏幕分辨率，旋转状态，亮灭屏状态等

  2）操作：按键，坐标操作，滑动，拖拽，灭屏唤醒屏幕，截图等

  3）监听器功能

单例模式

- getUiDevice() //调用时容易空指针
- UiDevice.*getInstance()* //推荐使用

按键与Keycode使用

手机常用按键

- HOME
- MENU
- BACK
- VOLUME_UP
- VOLUME_DOWN
- RecentApps
- POWER

按键API

返回值	方法名	描述
boolean	pressBack()	模拟短按Back键
boolean	pressDelete()	模拟短按Delete键
boolean	pressEnter()	模拟短按Enter键
boolean	pressHome()	模拟短按Home键
boolean	pressMenu()	模拟短按Menu键
boolean	pressRecentApps()	模拟短按最近使用程序
boolean	pressSearch()	模拟短按搜索键
boolean	pressKeyCode(int keyCode)	模拟短按键盘代码keyCode
boolean	pressKeyCode(int keyCode, int metaState)	模拟短按键盘代码keyCode

示例

```
public void testPress(){
try {
UiDevice.getInstance().pressRecentApps();
sleep(1000);
UiDevice.getInstance().pressHome();
} catch (RemoteException e) {
e.printStackTrace();
}
}
```

KEYCODE - 键盘映射码

- KeyEvent 按键事件
- META Key

  辅助功能键： ALT, SHIFT, Caps Lock

列	激活状态	metaState
base	META_key未被激活	0
caps	SHIFT或CAPS_LOCK被激活时	1
fn	ALT被激活	2
caps_fn	ALT, SHIFT或CAPS_LOCK同时被激活时	3

示例

```
public void testPressKeycode(){
UiDevice.getInstance().pressKeyCode(KeyEvent.KEYCODE_A);
UiDevice.getInstance().pressKeyCode(KeyEvent.KEYCODE_B);
UiDevice.getInstance().pressKeyCode(KeyEvent.KEYCODE_C);
UiDevice.getInstance().pressKeyCode(KeyEvent.KEYCODE_A,1);
UiDevice.getInstance().pressKeyCode(KeyEvent.KEYCODE_B,1);
UiDevice.getInstance().pressKeyCode(KeyEvent.KEYCODE_C,1);
}
```

获取坐标与坐标点击

坐标相关知识

- 手机屏幕坐标：从左上角开始到右下角
- DP：设备独立像素
- Point：代表一个点（x, y）
- Rect：矩形区域

示例

```
public void testRect() throws UiObjectNotFoundException{
//获取应用图标矩形区域
UiObject weChat=new UiObject(new UiSelector().description("微信"));
Rect rect=weChat.getBounds();
System.out.println(rect);
}
```

坐标相关API

返回值	方法名	描述
boolean	click(int x, int y)	使用坐标点击屏幕
int	getDisplayHeight()	获取屏幕高度
int	getDisplayWidth()	获取屏幕宽度
Point	getDisplaySizeDp()	获取显示尺寸返回显示大小

示例

```
public void testClick(){
//UiDevice.getInstance().click(161, 1000);
UiDevice.getInstance().click(161, 1646);

}
```

拖拽与滑动

拖拽与滑动相关知识

- 拖拽：将一个组件从一个坐标移动到另一个坐标
- 滑动：从一个坐标点移动到另一个坐标点
- 步长：从一点滑动到另一点使用的时间（1步长为5ms）

拖拽与滑动相关API

返回值	方法名	描述
boolean	drag(int startX, int startY, int endX, int endY, int steps)	拖到对象从一个坐标到另一个坐标
boolean	swipe(Point[] segments, int segmentSteps)	在点阵中滑动，5ms一步
boolean	swipe(int startX, int startY, int endX, int endY, int steps)	通过坐标滑动屏幕

示例

```
public void testDragAndSwipe(){
//亮屏
try {
while(!UiDevice.getInstance().isScreenOn()){
UiDevice.getInstance().wakeUp();
}
} catch (RemoteException e) {
e.printStackTrace();
}
//向上滑动解锁
UiDevice.getInstance().swipe(500, 1100, 500, 160, 20);
sleep(1000);
//拖拽
UiDevice.getInstance().drag(161, 1646, 161, 1000, 50);
}
```

示例

```
public void testSwipePoints(){
//密联图案解锁
Point p1=new Point(244,656);
Point p2=new Point(544,656);
Point p3=new Point(831,656);
Point p4=new Point(831,961);
Point[] pp={p1,p2,p3,p4};
UiDevice.getInstance().swipe(pp, 40);
}
```

旋转屏幕

旋转屏幕相关知识

- 旋转方向：4个方向，分别是0度，90度，180度，270度
- 重力感应器
- 固定位置与物理旋转

旋转屏幕相关API

返回值	方法名	描述
void	setOrientationLeft()	通过禁用传感器，然后模拟设备向左转，并且固定位置
void	setOrientationRight()	通过禁用传感器，然后模拟设备向右转，并且固定位置
void	setOrientationNatural()	通过禁用传感器，然后模拟设备转到其自然默认的位置，并且固定位置
void	unfreezeRotation()	重新启用传感器，并且允许物理旋转
boolean	isNaturalOrientation()	检测设备是否处于默认旋转状态
int	getDisplayRotation()	返回当前的显示旋转，0度，90度，180度，270度值分别为：0,1,2,3
void	freezeRotation()	禁用传感器，冻结设备旋转在当前的旋转状态

示例

```
public void testOrientation(){
//浏览器界面
try {
UiDevice.getInstance().setOrientationLeft();
sleep(1000);
if(!UiDevice.getInstance().isNaturalOrientation()){
UiDevice.getInstance().setOrientationNatural();
}
} catch (RemoteException e) {
e.printStackTrace();
}
}
```

灭屏与唤醒屏幕

灭屏与唤醒屏幕相关知识

- 灭屏：按电源键将屏幕熄灭
- 唤醒屏幕：在灭屏状态下按电源键点亮屏幕

灭屏与唤醒屏幕相关API

返回值	方法名	描述
void	wakeUp()	模拟按电源键，如果屏幕是唤醒的则没有任何作用
void	sleep()	模拟按电源键，如果屏幕是已经关闭的则没有任何作用
boolean	isScreenOn()	检查屏幕是否亮屏

示例

```
public void testWakeupAndSleep(){
try {
if(!UiDevice.getInstance().isScreenOn()){
UiDevice.getInstance().wakeUp();
}else{
UiDevice.getInstance().sleep();
}
} catch (RemoteException e) {
e.printStackTrace();
}
}
```

截图与等待空闲

截图与等待空闲相关知识

- 图片缩放比例
- 图片质量
- File类
- 图片格式：PNG
- 空闲状态
- 窗口更新事件

截图相关API

返回值	方法名	描述
boolean	takeScreenshot(storePath)	把当前窗口截图并将其存储为PNG格式，默认为1.0f大小（原尺寸）和90%质量。 参数为file类的文件路径
boolean	takeScreenshot(storePath, scale, quality)	把当前窗口截图并将其存储为PNG格式，可以自定义缩放比例和质量

参数说明：

storePath：存储路径，必须为png格式

scale：缩放比例，1.0为原图

quality：图片压缩质量，范围为0-100

等待空闲相关API

返回值	方法名	描述
void	waitForIdle()	等待当前应用处于空闲状态，默认等待10s
void	waitForIdle(timeout)	自定义超时等待当前应用处于空闲状态
boolean	waitForWindowUpdate(packageName, timeout)	等待窗口内容更新事件的发生

示例

```
public void testScreenShot(){
UiDevice.getInstance().takeScreenshot(new
File("/sdcard/1.png"));
}
```

获取包名、开启通知栏、快速设置、获取布局文件的方法

相关知识

- 包名
- 通知栏
- 快速设置
- 布局文件

包名、通知栏、快速设置、布局文件相关API

返回值	方法名	描述
void	getCurrentPackageName()	获取当前界面的包名
boolean	openNotification()	打开通知栏
boolean	openQuickSettings()	打开快速设置
void	dumpWindowHierarchy(fileName)	获取当前界面的布局文件，保存在/data/local/tmp目录下

示例

```
public void testGetPackageName(){
String packageName = UiDevice.getInstance().getProductName();
System.out.println("packageName:"+packageName);
UiDevice.getInstance().openNotification();
sleep(2000);
UiDevice.getInstance().openQuickSettings();
}
```

实例演示

示例：打开浏览器，输入百度网址

- 步骤

灭屏->亮屏->解锁->单击浏览器->单击网址输入框->输入www.baidu.com->按回车键->旋转屏幕->截图

示例

```
public void testDemo() throws RemoteException{
//灭屏-》亮屏->解锁->单击浏览器->单击网址输入框->输入www.baidu.com->按回车键->旋转屏幕->截图
UiDevice.getInstance().sleep();
sleep(1500);
if(!UiDevice.getInstance().isScreenOn()){
UiDevice.getInstance().wakeUp();
}
sleep(1500);
//向上滑动解锁
UiDevice.getInstance().swipe(500, 1100, 500, 160, 20);
UiDevice.getInstance().pressHome();
sleep(1500);
UiDevice.getInstance().click(728, 1640);
sleep(2000);
UiDevice.getInstance().click(172, 158);
sleep(1500);
UiDevice.getInstance().pressDelete();
sleep(1500);
UiDevice.getInstance().pressKeyCode(KeyEvent.KEYCODE_W);
sleep(500);
UiDevice.getInstance().pressKeyCode(KeyEvent.KEYCODE_W);
sleep(500);
```

```
UiDevice.getInstance().pressKeyCode(KeyEvent.KEYCODE_W);
sleep(500);
UiDevice.getInstance().pressKeyCode(KeyEvent.KEYCODE_PERIOD);
sleep(500);
UiDevice.getInstance().pressKeyCode(KeyEvent.KEYCODE_B);
sleep(500);
UiDevice.getInstance().pressKeyCode(KeyEvent.KEYCODE_A);
sleep(500);
UiDevice.getInstance().pressKeyCode(KeyEvent.KEYCODE_I);
sleep(500);
UiDevice.getInstance().pressKeyCode(KeyEvent.KEYCODE_D);
sleep(500);
UiDevice.getInstance().pressKeyCode(KeyEvent.KEYCODE_U);
sleep(500);
UiDevice.getInstance().pressKeyCode(KeyEvent.KEYCODE_PERIOD);
sleep(500);
UiDevice.getInstance().pressKeyCode(KeyEvent.KEYCODE_C);
sleep(500);
UiDevice.getInstance().pressKeyCode(KeyEvent.KEYCODE_O);
sleep(500);
UiDevice.getInstance().pressKeyCode(KeyEvent.KEYCODE_M);
sleep(500);
UiDevice.getInstance().pressEnter();
sleep(2000);
UiDevice.getInstance().setOrientationLeft();
sleep(1500);
UiDevice.getInstance().takeScreenshot(new File("/sdcard/2.png"));
}
```

Q&A