

UiAutomator UiScrollable类API (五)

By Zhang Jingwei

20160308

概要

- UiScrollable类介绍
- 快速滚动
- 获取列表子元素
- 获取与设置最大滚动次数常量
- 获取与设置滑动区域校准常量
- 向前与向后滚动
- 滚动到某个对象
- 设置滚动方向

UIScrollView类介绍

UIScrollView类说明

- UIScrollView是UICollectionView的子类
- UIScrollView专门处理滚动事件，提供各种滚动方法

快速滚动	向前滚动	向后滚动
滚动到某个对象	设置滚动方向	设置滚动次数

快速滚动

1.滚动相关

- 步长
- 扫动次数

快速滚动

2.快速滚动相关API

返回值	方法名	描述
boolean	<code>flingBackward()</code>	快速向后滑动，步长为5
boolean	<code>flingForward()</code>	快速向前滑动，步长为5
boolean	<code>flingToBeginning(int <u>maxSwipes</u>)</code>	自定义扫动次数快速滑动到开始，步长为5
boolean	<code>flingToEnd(int <u>maxSwipes</u>)</code>	自定义扫动次数快速滑动到结束，步长为5

示例

```
public void testFling() throws UiObjectNotFoundException  
//WeChat Region,快速滚动  
UiScrollable regionScr=new UiScrollable(new  
UiSelector().classNameMatches(".*ListView"));  
regionScr.flingForward();  
sleep(3000);  
regionScr.flingBackward();  
sleep(3000);  
regionScr.flingToBeginning(10);  
sleep(3000);  
regionScr.flingToEnd(10);  
}
```

获取列表子元素

返回值	方法名	描述
UiObject	<code>getChildByDescription(UiSelector <u>childPattern</u>, String text)</code>	默认滚动获取具备UiSelector条件的元素集合后再以描述条件查找对象
UiObject	<code>getChildByDescription(UiSelector <u>childPattern</u>, String text, <u>boolean allowScrollSearch</u>)</code>	是否允许滚动获取具备UiSelector条件的元素集合后再以描述条件查找对象
UiObject	<code>getChildByInstance(UiSelector <u>childPattern</u>, int instance)</code>	获取具备UiSelector条件的元素集合后再按实例查找对象（不滚动）
UiObject	<code>getChildByText(UiSelector <u>childPattern</u>, String text)</code>	默认滚动获取具备UiSelector条件的元素集合后再以文本条件查找对象
UiObject	<code>getChildByText(UiSelector <u>childPattern</u>, String text, <u>boolean allowScrollSearch</u>)</code>	是否允许滚动获取具备UiSelector条件的元素集合后再以文本条件查找对象

示例

```
public void testGetChild() throws UiObjectNotFoundException{  
//WeChat Region,获取子元素  
UiScrollable regionScr=new UiScrollable(new  
UiSelector().classNameMatches(".*ListView"));  
UiObject regionChina=regionScr.getChildByText(new  
UiSelector().classNameMatches(".*TextView"), "China");  
regionChina.click();  
sleep(3000);  
UiObject regionBtn=new UiObject(new UiSelector().resourceIdMatches(".*id/lv"));  
regionBtn.clickAndWaitForNewWindow();  
sleep(1500);  
UiObject regionAlgeria=regionScr.getChildByInstance(new  
UiSelector().classNameMatches(".*TextView"), 3);  
regionAlgeria.click();  
}
```


获取与设置最大滚动次数常量

1. 相关API

返回值	方法名	描述
int	getMaxSearchSwipes()	获取执行搜索滑动过程中的最大滑动次数，默认为30
UiScrollable	setMaxSearchSwipes(int <u>swipes</u>)	设置最大滑动次数

示例

```
public void testMaxSearchSwipe() throws UiObjectNotFoundException{  
//WeChat Region,最大滑动次数  
UiScrollable regionScr=new UiScrollable(new  
UiSelector().classNameMatches(".*ListView"));  
int maxSearchSwipe=regionScr.getMaxSearchSwipes();  
System.out.println("MaxSearchSwipes::"+maxSearchSwipe);  
sleep(500);  
regionScr.setMaxSearchSwipes(35);  
int curMaxSearchSwipe=regionScr.getMaxSearchSwipes();  
System.out.println("CurMaxSearchSwipes::"+curMaxSearchSwipe);  
sleep(500);  
regionScr.setMaxSearchSwipes(30);  
}
```

获取与设置滑动区域校准常量

1.校准相关

- 滑动操作坐标时的偏移量，用来取偏移比例

获取与设置滑动区域校准常量

2. 相关API

返回值	方法名	描述
double	<code>getSwipeDeadZonePercentage()</code>	默认为0.1,10%
UiScrollable	<code>setSwipeDeadZonePercentage(double <u>swipeDeadZonePercentage</u>)</code>	设置一个部件的大小，在滑动时，视为无接触区的百分比

示例

```
public void testDeadZonePercentage() throws UiObjectNotFoundException{  
//WeChat Region,校准常量  
UiScrollable regionScr=new UiScrollable(new  
UiSelector().classNameMatches(".*ListView"));  
double swipeDeadZonePercentage=regionScr.getSwipeDeadZonePercentage();  
System.out.println("swipeDeadZonePercentage::"+swipeDeadZonePercentage);  
regionScr.flingForward();  
sleep(2000);  
regionScr.setSwipeDeadZonePercentage(0.45);  
double curSwipeDeadZonePercentage=regionScr.getSwipeDeadZonePercentage();  
System.out.println("curSwipeDeadZonePercentage::"+curSwipeDeadZonePercentage);  
sleep(500);  
regionScr.flingForward();  
sleep(500);  
regionScr.setSwipeDeadZonePercentage(0.10);  
}
```

向前与向后滚动

1. 相关API

返回值	方法名	描述
boolean	<code>scrollBackward()</code>	向后滚动，默认步长为55
boolean	<code>scrollBackward(int <u>steps</u>)</code>	向后滚动，自定义步长
boolean	<code>scrollForward()</code>	向前滚动，默认步长为55
boolean	<code>scrollForward(int <u>steps</u>)</code>	向前滚动，自定义步长

示例

```
public void testScrollBackForward() throws UiObjectNotFoundException{  
//WeChat Region,向前向后滚动  
UiScrollable regionScr=new UiScrollable(new  
UiSelector().classNameMatches(".*ListView"));  
regionScr.scrollBackward();//默认步长55  
sleep(2000);  
regionScr.scrollBackward(5);  
}
```

滚动到某个对象

1. 相关API

返回值	方法名	描述
boolean	<code>scrollDescriptionIntoView(String <u>text</u>)</code>	滚动到描述所在位置，并尽量让它居于屏幕中央
boolean	<code>scrollIntoView(UiObjector obj)</code>	滚动到对象所在位置，并尽量让它居于屏幕中央
boolean	<code>scrollIntoView(UiSelector <u>selector</u>)</code>	滚动到条件元素所在位置，并尽量让它居于屏幕中央
boolean	<code>scrollTextIntoView(String <u>text</u>)</code>	滚动到文本对象所在位置，并尽量让它居于屏幕中央

示例

```
public void testScrollIntoView() throws UiObjectNotFoundException  
//RE文件管理器,滚动到某个对象  
UiScrollable appScr=new UiScrollable(new UiSelector().classNameMatches(".*ListView"));  
UiSelector selector=new UiSelector().text("root");  
UiObject dirRoot=new UiObject(selector);  
appScr.scrollToView(dirRoot);  
dirRoot.click();  
sleep(2000);  
UiDevice.getInstance().pressBack();  
sleep(1000);  
UiObject dirData=new UiObject(new UiSelector().text("data"));  
appScr.scrollToView("data");  
dirData.click();  
}
```

滚动到某个对象

2. 相关API

返回值	方法名	描述
boolean	<code>scrollToBeginning(int <u>maxSwipes</u>)</code>	滚动到开始位置
boolean	<code>scrollToBeginning(int <u>maxSwipes</u>, <u>int steps</u>)</code>	滚动到开始位置，自定义扫动距离与步长
boolean	<code>scrollToEnd(int <u>maxSwipes</u>)</code>	滚动到结束位置
boolean	<code>scrollToEnd(int maxSwipes, int steps)</code>	滚动到结束位置，自定义扫动距离与步长

示例

```
public void testScrollToBeginEnd() throws UiObjectNotFoundException{  
//WeChat Region,滚动到开始、结束  
UiScrollable regionScr=new UiScrollable(new  
UiSelector().classNameMatches(".*ListView"));  
regionScr.scrollToEnd(5);  
sleep(2000);  
regionScr.scrollToBeginning(20, 5);  
}
```

设置滚动方向

1. 相关API

返回值	方法名	描述
UiScrollable	setAsHorizontalList()	设置滚动方向为水平滚动
UiScrollable	setAsVerticalList()	设置滚动方向为纵向滚动

示例

```
public void testSetHorVer() throws UiObjectNotFoundException{  
//联系人,滚动方向  
UiScrollable contactScr=new UiScrollable(new  
UiSelector().classNameMatches(".*ViewPager"));  
contactScr.setAsVerticalList();  
sleep(1000);  
contactScr.scrollForward();  
sleep(2000);  
contactScr.setAsHorizontalList();  
contactScr.scrollBackward();  
}
```

实例演示

示例：打开联系人，拨打电话
步骤

灭屏->亮屏->单击电话->滚动到联系人列表
（水平）->滚动到指定联系人（纵向）->点击联系人
->拨打10010->接通几秒，挂断电话->退出

示例

```
public void testDemo() throws UiObjectNotFoundException, RemoteException{
```

```
//拨打10010
```

```
//灭屏->亮屏->单击电话->滚动到联系人列表（水平）->滚动到指定联系人（纵向）->  
单击联系人->拨打10010->接通几秒，挂断电话->退出
```

```
UiDevice.getInstance().sleep();
```

```
sleep(1500);
```

```
if(!UiDevice.getInstance().isScreenOn()){
```

```
UiDevice.getInstance().wakeUp();
```

```
}
```

```
sleep(500);
```

```
UiDevice.getInstance().pressHome();
```

```
sleep(1000);
```

```
UiObject phoneBtn=new UiObject(new UiSelector().textContains("Phone"));
```

```
phoneBtn.clickAndWaitForNewWindow();
```

```
sleep(500);
```

```
UiScrollable contactScr=new UiScrollable(new  
UiSelector().classNameMatches(".*ViewPager")));  
contactScr.setAsHorizontalList();  
contactScr.scrollForward();  
contactScr.scrollForward();  
sleep(1000);  
contactScr.setAsVerticalList();  
contactScr.scrollToTextIntoView("Zjw");  
UiObject zjwBtn=new UiObject(new UiSelector().text("Zjw")));  
zjwBtn.clickAndWaitForNewWindow();  
UiObject n10010Btn=new UiObject(new UiSelector().text("10010")));  
sleep(1000);  
n10010Btn.click();  
sleep(10000);  
UiDevice.getInstance().pressKeyCode(KeyEvent.KEYCODE_ENDCALL);  
sleep(1000);  
UiDevice.getInstance().pressHome();  
}
```


Q&A