

UiAutomator UiSelector类API (四)

By Zhang Jingwei

20160301

概要

- UiSelector类介绍
- Android常用布局与常用组件
- 四种匹配关系
- 节点关系
- 对象定位-文本与描述
- 对象定位-类名与包名
- 对象定位-索引与实例
- 对象定位-特殊属性与节点
- 对象定位-Resource ID

UiSelector类介绍

- UiSelector类说明
- UiSelector功能
 - 通过各种属性与节点关系定位组件
- 自动化操作步骤
 - 找到对象->操作对象

Android常用布局与常用组件

1.常用布局

- 线性布局 `LinearLayout`
- 相对布局 `RelativeLayout`
- 表格布局 `TableLayout`
- 帧布局 `RelativeLayout`
- 网格布局 `RelativeLayout`
- 绝对布局 `AbsoluteLayout`

2.常用组件

- TextView
- EditText
- Button
- RadioButton
- CheckBox
- ListView
- ProgressBar
- Toast
- ScrollView
- ...

四种匹配关系

1. 四种匹配关系

- 完全匹配（默认）
- 包含匹配（Contains）
- 正则匹配（Matches）- 完全匹配、包含匹配、起始匹配
- 起始匹配（StartsWith）

2. 示例

匹配字符串：123456

- 完全匹配：123456
- 包含匹配：234
- 正则匹配：\d{6}
- 起始匹配：123

节点关系

- 父 Parent
- 子 Children
- 同胞 Sibling

-
(0) RelativeLayout [810,1628][1080,1712]

(0) ImageView [897,1628][993,1712]

(1) ImageView [971,1631][1001,1661]

对象定位-文本与描述

1. 文本属性定位对象

返回值	方法名	描述
UiSeletor	text(String <u>text</u>)	文本
UiSelector	textContains(String <u>text</u>)	文本包含
UiSelector	textMatches(String <u>regex</u>)	文本正则
UiSelector	textStartsWith(String <u>text</u>)	文本起始匹配

示例

```
public void testByText() throws UiObjectNotFoundException{  
//微信好友列表  
UiObject newfrindsBtn=new UiObject(new UiSelector().text("新的朋友"));//text  
newfrindsBtn.click();  
sleep(1500);  
UiDevice.getInstance().pressBack();  
UiObject newfrindsBtn2=new UiObject(new UiSelector().textContains("朋友"));//textContains  
newfrindsBtn2.click();  
sleep(1500);  
UiDevice.getInstance().pressBack();  
UiObject newfrindsBtn3=new UiObject(new UiSelector().textMatches("..\\u670b*."));//textMatches  
newfrindsBtn3.click();  
sleep(1500);  
UiDevice.getInstance().pressBack();  
UiObject newfrindsBtn4=new UiObject(new UiSelector().textStartsWith("新的"));//textStartsWith  
newfrindsBtn4.click();  
UiDevice.getInstance().pressBack();  
}
```

对象定位-文本与描述

2.描述属性定位对象

返回值	方法名	描述
UiSelector	description(String <u>desc</u>)	描述
UiSelector	descriptionContains(String <u>desc</u>)	描述包含
UiSelector	descriptionMatches(String <u>regex</u>)	描述正则
UiSelector	descriptionStartsWith(String <u>desc</u>)	描述起始匹配

示例

```
public void testByDesc() throws UiObjectNotFoundException{  
//微信好友列表  
UiObject yueYueBtn=new UiObject(new UiSelector().description("A6217月月"));  
yueYueBtn.click();//description  
sleep(1500);  
UiDevice.getInstance().pressBack();  
UiObject yueYueBtn2=new UiObject(new UiSelector().descriptionContains("月月"));  
yueYueBtn2.click();//descriptionContains  
sleep(1500);  
UiDevice.getInstance().pressBack();  
UiObject yueYueBtn3=new UiObject(new UiSelector().descriptionMatches(".6217.."));  
yueYueBtn3.click();//descriptionMatches  
sleep(1500);  
UiDevice.getInstance().pressBack();  
UiObject yueYueBtn4=new UiObject(new UiSelector().descriptionStartsWith("A62"));  
yueYueBtn4.click();//descriptionStartsWith  
sleep(1500);  
UiDevice.getInstance().pressBack();  
}
```

对象定位-类名与包名

1.类名属性定位对象

返回值	方法名	描述
UiSeletor	className(String <u>className</u>)	类名
UiSelector	classNameMatches(String <u>regex</u>)	类名正则

示例

```
public void testByClassName() throws UiObjectNotFoundException{  
//微信聊天列表  
UiObject pandaBtn=new UiObject(new UiSelector()  
.className("android.widget.LinearLayout").index(11));//className  
pandaBtn.click();  
sleep(1500);  
UiDevice.getInstance().pressBack();  
UiObject pandaBtn2=new UiObject(new UiSelector()  
.classNameMatches(".*LinearLayout").index(11));//classNameMatches  
pandaBtn2.click();  
sleep(1500);  
UiDevice.getInstance().pressBack();  
}
```

对象定位-类名与包名

2.包名属性定位对象

返回值	方法名	描述
UiSeletor	packageName(String <u>name</u>)	包名
UiSelector	packageNameMatches(String <u>regex</u>)	包名正则

示例

```
public void testByPkgName() throws UiObjectNotFoundException{  
//微信聊天列表  
UiObject pandaBtn=new UiObject(new UiSelector()  
.packageName("com.tencent.mm").textContains("A76"));//packageName  
pandaBtn.click();  
sleep(1500);  
UiDevice.getInstance().pressBack();  
UiObject pandaBtn2=new UiObject(new UiSelector()  
.packageNameMatches(".*cent.mm").textContains("A76"));//packageNameMatches  
pandaBtn2.click();  
sleep(1500);  
UiDevice.getInstance().pressBack();  
}
```

对象定位-索引与实例

1.索引与实例

- 索引 `index` (同一层级)
- 实例 `instance`(整个布局文件)

对象定位-索引与实例

2.索引与实例属性定位对象

返回值	方法名	描述
UiSeletor	index(int <u>index</u>)	索引
UiSelector	instance(int <u>instance</u>)	实例

示例

```
public void testByIndexAndInstance() throws UiObjectNotFoundException{  
//计算器  
UiObject eightBtn=new UiObject(new UiSelector()  
    .className("android.widget.Button").index(1));  
eightBtn.click();  
sleep(1500);  
UiDevice.getInstance().pressDelete();  
UiObject eightBtn2=new UiObject(new UiSelector()  
    .className("android.widget.Button").instance(1));  
eightBtn2.click();  
sleep(1500);  
UiDevice.getInstance().pressDelete();  
}
```

对象定位-特殊属性与节点

1.特殊属性定位对象

返回值	方法名	描述
UiSelector	checked(boolean val)	checked属性
UiSelector	clickable(boolean val)	可点击属性
UiSelector	enabled(boolean val)	enabled属性
UiSelector	focusable(boolean val)	焦点属性
UiSelector	focused(boolean val)	当前焦点属性
UiSelector	longClickable(boolean val)	长按属性
UiSelector	scrollable(boolean val)	滚动属性
UiSelector	selected(boolean val)	selected属性

示例

```
public void testByCheckable() throws UiObjectNotFoundException{  
//设置蓝牙  
UiObject btBtn=new UiObject(new UiSelector()  
.checkable(true));//checkable  
if(!btBtn.isChecked()){  
    btBtn.click();  
    sleep(1500);  
}else{  
    btBtn.click();  
}  
}
```

对象定位-特殊属性与节点

2.节点属性定位对象

返回值	方法名	描述
UiSeletor	childSelector(UiSelector <u>selector</u>)	子类
UiSelector	fromParent(UiSelector <u>selector</u>)	父类

示例

```
public void testNode() throws UiObjectNotFoundException{  
//计算器  
UiObject childNine=new UiObject(new UiSelector()  
.className("android.view.View").childSelector(new UiSelector().index(2))));  
childNine.click();//childSelector  
sleep(1500);  
UiDevice.getInstance().pressDelete();  
}
```

对象定位-Resource ID

1. Resource ID

- Resource ID属性
- 对应Android属性(layout下xml文件里的android:id)

对象定位-Resource ID

2.Resource ID属性定位对象

返回值	方法名	描述
UiSeletor	resourceId(<u>String id</u>)	Resource ID
UiSelector	resourceIdMatches(<u>String regex</u>)	Resource ID正则

示例

```
public void testByResId() throws UiObjectNotFoundException{  
//微信聊天窗口  
UiObject iconBtn=new UiObject(new UiSelector()  
    .resourceId("com.tencent.mm:id/wi"));  
iconBtn.click();//resourceId  
sleep(1500);  
UiDevice.getInstance().pressBack();  
sleep(1500);  
UiObject iconBtn2=new UiObject(new UiSelector()  
    .resourceIdMatches(".*id/wi"));  
iconBtn2.click();//resourceIdMatches  
sleep(1500);  
}
```

实例演示

示例：打开菜鸟商店，安装万年历
步骤

灭屏->亮屏->解锁->单击菜鸟商店->单击万年历->点击下载->点击确定->打开通知列表->等待下载完成，点击->点击安装->等待安装完成->点击完成->退出

示例

```
public void testDemo() throws UiObjectNotFoundException, RemoteException{  
//菜鸟商店  
//灭屏->亮屏->解锁->单击菜鸟商店->单击万年历->点击下载->点击确定->打开通知  
//列表->等待下载完成, 点击->点击安装->等待安装完成->点击完成->退出  
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);  
UiObject appstoreBtn=new UiObject(new UiSelector().textContains("菜鸟"));  
appstoreBtn.clickAndWaitForNewWindow();  
UiObject wanNl=new UiObject(new  
UiSelector().resourceIdMatches(".*id/appname").textMatches(".*万年历"));  
wanNl.click();  
sleep(1500);  
}
```

```
UiObject downLoad=new UiObject(new
UiSelector().classNameMatches(".*ImageView").instance(1));
downLoad.clickAndWaitForNewWindow();
sleep(1500);
UiObject okBtn=new UiObject(new
UiSelector().resourceIdMatches(".*id/button_primary");
okBtn.click();
sleep(1500);
UiDevice.getInstance().openNotification();
sleep(1500);
UiObject finishBtn=new UiObject(new UiSelector().textContains("已完成");
finishBtn.waitForExists(15000);
finishBtn.clickAndWaitForNewWindow();
UiObject installBtn=new UiObject(new UiSelector().text("安装");
installBtn.click();
UiObject finishInsBtn=new UiObject(new
UiSelector().classNameMatches(".*Button").text("完成");
finishInsBtn.waitForExists(15000);
sleep(1000);
finishInsBtn.click();
sleep(200);
UiDevice.getInstance().pressHome();
}
```

Q&A