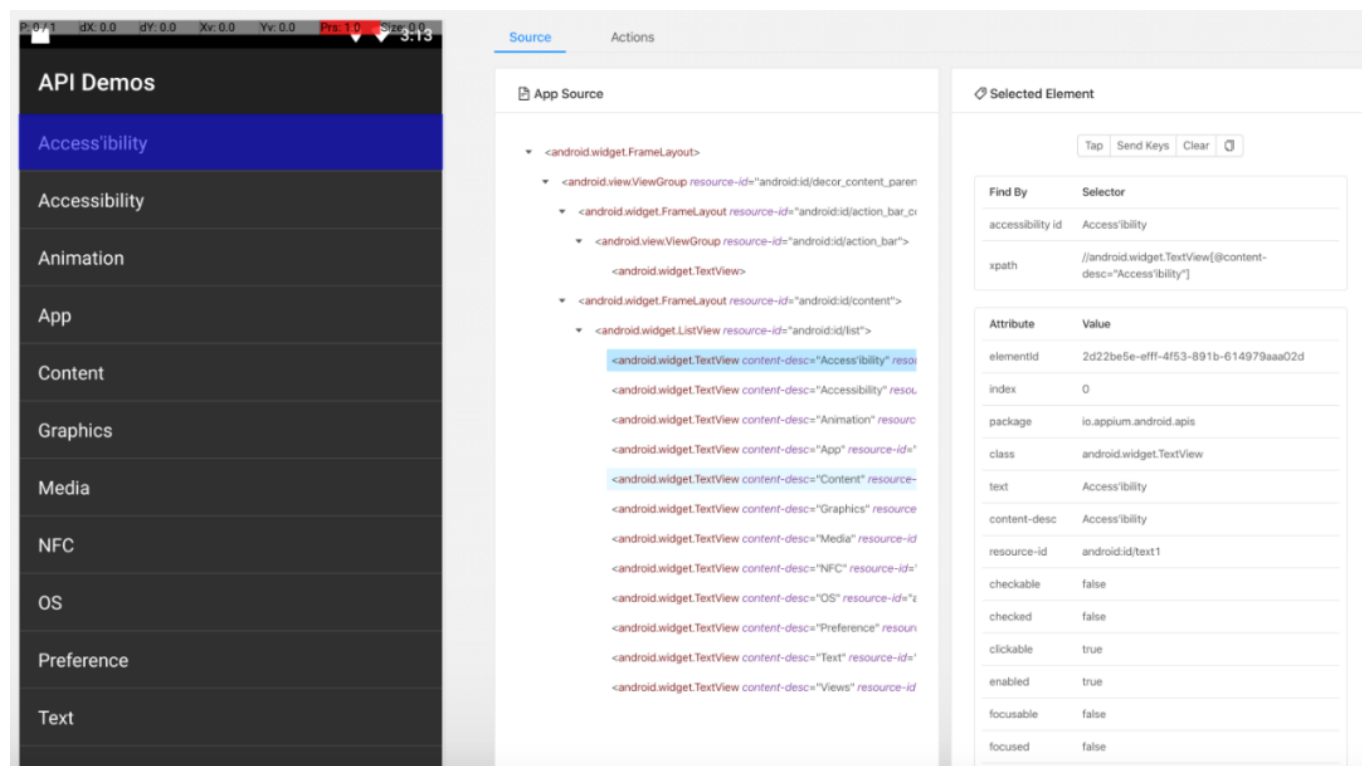


客户端的页面通过 XML 来实现 UI 的布局，页面的 UI 布局作为一个树形结构，而树叶被定义为节点。这里的节点也就对应了要定位的元素，节点的上级节点，定义了元素的布局结构。在 XML 布局中可以使用 XPath 进行节点的定位。

App的布局结构



从上面这张图中可以看到最左侧是应用的页面的展示，中间部分展示了这个页面的树形结构的 XML 代码。

其中包含的内容为：

- 节点 node
- 节点属性：包括 clickable（是否可点击）、content-desc（内容）、resource-id（元素 id）、text（文本）、bounds（坐标）等。

通过 ID 定位

在 Android 系统元素的 ID 称为 resource-id，使用页面分析工具比如 Appium Inspector 能够获取元素的唯一标识是 ID 属性，可以使用 ID 进行元素定位，方便快捷。

示例代码如下：

- Python 版本

```
1 driver.find_element(By.ID, "android:id/text1").click()
```

- Java 版本

```
1 driver.findElement(By.id("android:id/text1")).click();
```

注意 resource-id 对应的属性（包名:id/id 值），在使用这个属性的时候要把它当作一个整体。

通过 Accessibility 定位

当分析工具能抓取到的 content-desc 的属性值是唯一时，可以采用 Accessibility 的定位方式，示例代码：

- Python 版本

```
1 driver.find_element_by_accessibility_id("Accessibility")
```

- Java 版本

```
1 driver.findElementByAccessibilityId("Accessibility");
```

通过 XPath 定位

与 Selenium 类似，可以使用 XPath 的定位方式完成页面的元素定位。XPath 分为绝对路径定位与相对路径定位两种形式，下面介绍的都是相对定位的形式。

XPath：resource-id 属性定位

元素可以通过 resource-id 定位。

格式：

```
1 //*[@resource-id='resource-id属性']
```

示例代码：

- Python 版本

```
1 driver.find_element(By.XPATH, \'//*[@resource-id="rl_login_phone"]\')
```

- Java 版本

```
1 driver.findElement(By.xpath(\'//*[@resource-id="rl_login_phone"]\'));
```

XPath：text 属性定位

元素可以通过 text 文本属性定位。

格式：

```
1 //*[@text='text文本属性']
```

示例代码如下

- Python 版本

```
1 driver.find_element(By.XPATH, '//*[@text="我的"]')
```

- Java 版本

```
1 driver.findElement(By.xpath("//*[@text=\"我的\"]"));
```

XPath : class 属性定位

元素可以通过 class 定位。

格式：

```
1 //*[@class='class 属性']
```

示例代码：

- Python 版本

```
1 driver.find_element(By.XPATH, '//*[@class="android.widget.EditText"]')
```

- Java 版本

```
1 driver.findElement(By.xpath("//*[@class=\"android.widget.EditText\"]"));
2
```

XPath : content-desc 属性定位

元素可以通过 content-desc 定位。

格式：

```
1 //*[@content-desc='content-desc 属性']
```

示例代码：

- Python 版本

```
1 driver.find_element((By.XPATH, '//*[@content-desc="搜索"]'))
```

- Java 版本

```
1
2 driver.findElement(By.xpath("//*[@content-desc=\"搜索\"]"));
3
```

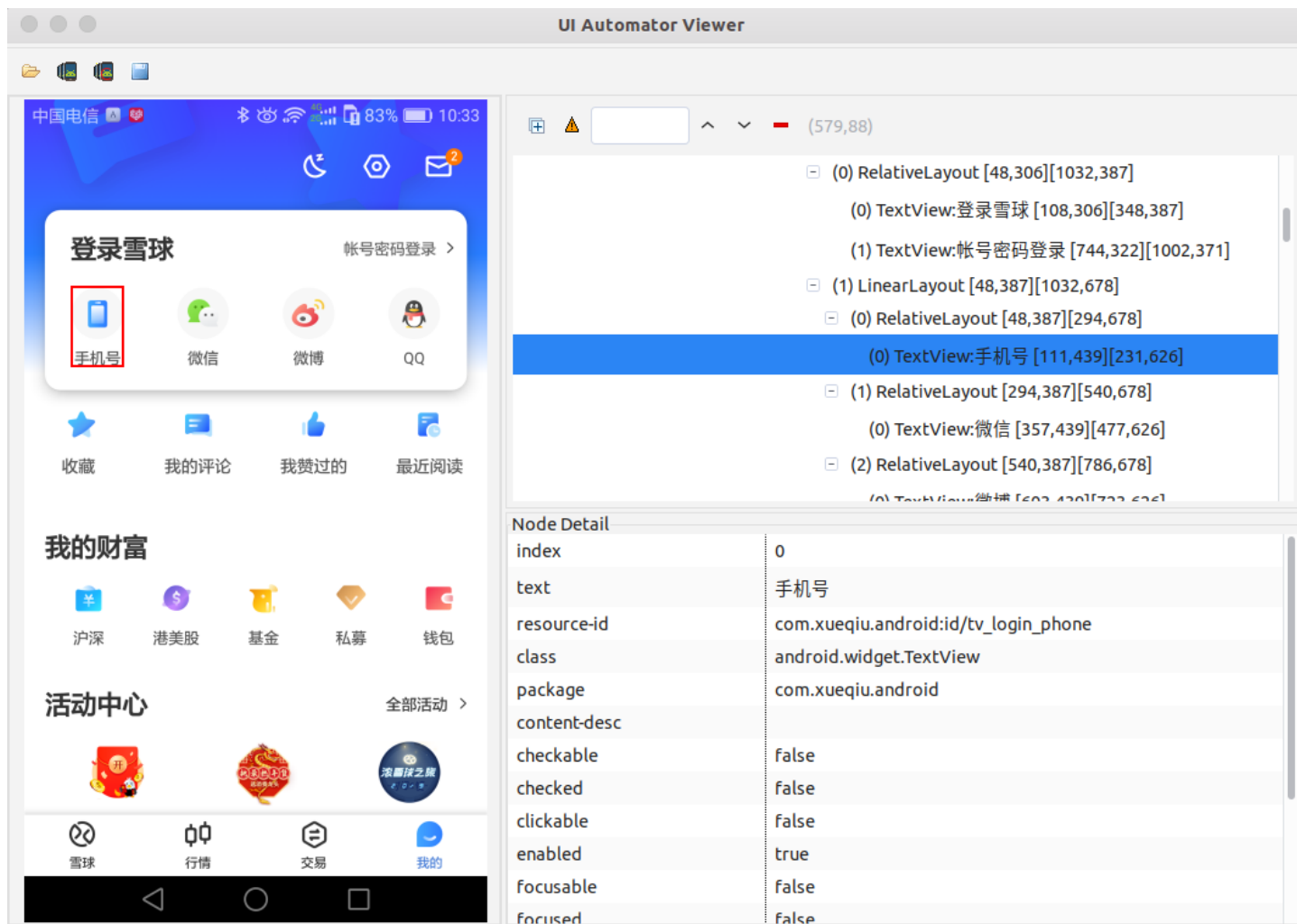
uiautomatorviewer介绍

使用 Android SDK (sdk/tools/uiautomatorviewer) 路径下自带的 uiautomatorviewer 工具也可以抓取当前页面的元素。

提前配置 `sdk/tools/` 路径到环境变量 `$PATH` 中，直接在命令行输入下面的命令：

```
1 uiautomatorviewer
```

可以打开下面这样一个页面，点击页面左上角第二个图标（Android 手机图标），就可以获取下面的 uiautomatorviewer 快照图：



uiautomatorviewer 抓取快照展示出来的元素属性是经过解析的，如果想要查看 XML DOM 的真实结构可以打印 pagesource，得到的内容如下，红色框起来的部分为上图的定位的 XML DOM 中的一个节点：

```
<android.widget.RelativeLayout index="0" package="com.xueqiu.android"
class="android.widget.RelativeLayout" text="" resource-id="com.xueqiu.android:id/rl_login_phone"
checkable="false" checked="false" clickable="true" enabled="true" focusable="true" focused="false" long-
clickable="false" password="false" scrollable="false" selected="false" bounds="[48,385][294,687]"
displayed="true">
    <android.widget.TextView index="0" package="com.xueqiu.android"
class="android.widget.TextView" text="手机号" resource-id="com.xueqiu.android:id/tv_login_phone"
checkable="false" checked="false" clickable="false" enabled="true" focusable="false" focused="false" long-
clickable="false" password="false" scrollable="false" selected="false" bounds="[111,446][231,626]"
displayed="true" />
</android.widget.RelativeLayout>
<android.widget.RelativeLayout index="1" package="com.xueqiu.android"
class="android.widget.RelativeLayout" text="" resource-id="com.xueqiu.android:id/rl_login_wx"
checkable="false" checked="false" clickable="true" enabled="true" focusable="true" focused="false" long-
clickable="false" password="false" scrollable="false" selected="false" bounds="[294,385][540,687]"
displayed="true">
```

通过图片分析，`android.widget.TextView` 是文本类型的节点，其中包含的属性信息都在上面的 uiautomatorviewer 快照图中有展示。如果只想定位 Android 系统的页面元素，可以直接使用

uiautomatorviewer，速度快并且不需要配置任何参数，直接点击获取页面的图标就可以将客户端页面抓取出来。

另外，uiautomatorviewer 只能抓取 android8 以下的版本，如果要抓取 android8 以上的版本的页面信息，可以使用 Appium Inspector 或 WEditor。