Technology Tarena 达内科技

使用Python语言实现Appium自动化测试

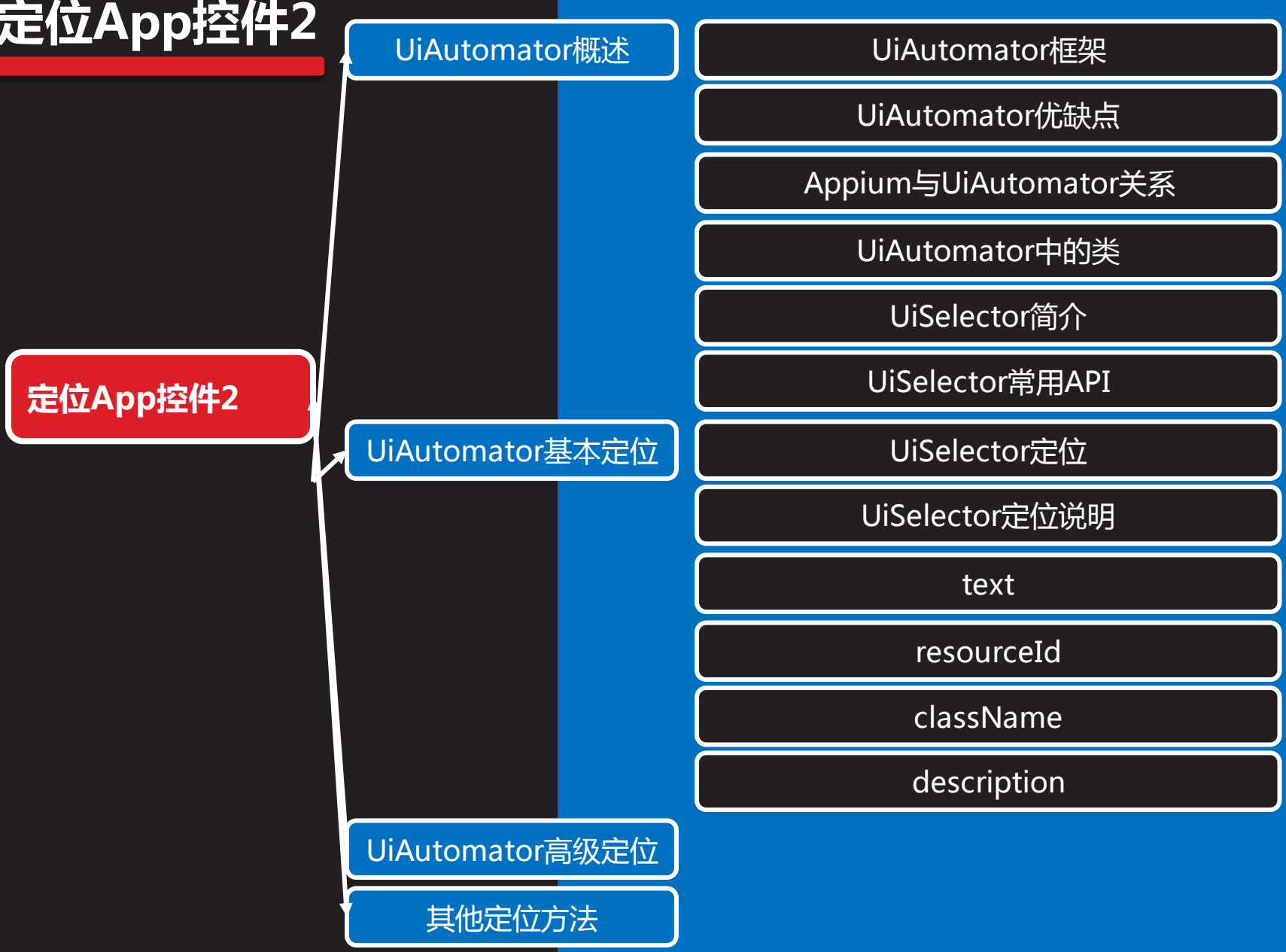
APPIUM WITH PYTHON

DAY06

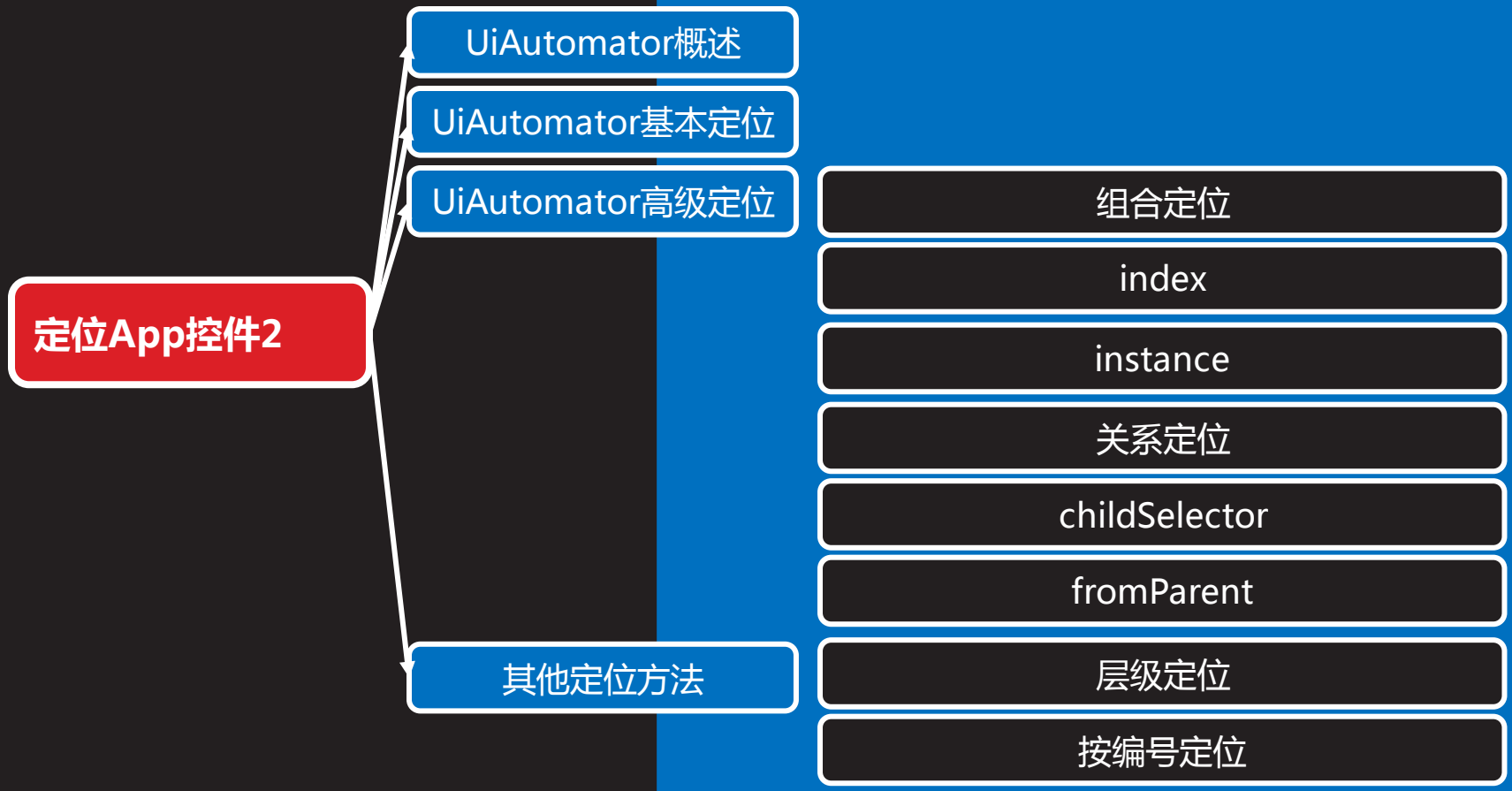
内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	定位App控件2
	10:30 ~ 11:20	
	11:30 ~ 12:20	
下午	14:00 ~ 14:50	
	15:00 ~ 15:50	
	16:00 ~ 16:50	
	17:00 ~ 17:30	总结和答疑

定位App控件2


定位App控件2
UiAutomator概述
UiAutomator框架
UiAutomator优缺点
Appium与UiAutomator关系
UiAutomator中的类
UiSelector简介
UiSelector常用API
UiAutomator基本定位
UiSelector定位
UiSelector定位说明
text
resourceId
className
description
UiAutomator高级定位
其他定位方法

定位App控件2（续1）


定位App控件2
UiAutomator概述
UiAutomator基本定位
UiAutomator高级定位
其他定位方法
组合定位
index
instance
关系定位
childSelector
fromParent
层级定位
按编号定位

Technology Tarena 达内科技

UiAutomator概述

UiAutomator框架

- UiAutomator是Google参考微软的UiAutomation提供的一套用在Android上的自动化测试框架。
- 是谷歌在Android4.1版本发布时推出的一款用Java编写的Ui测试框架，所以UiAutomator只能运行在4.1以后的版本中。
- 它只能用于UI也就是黑盒方面的测试。
- Uiautomator框架提供的API对安卓应用进行操作，如点击、滑动、键盘输入、长按以及常用的断言方法等。

知识讲解

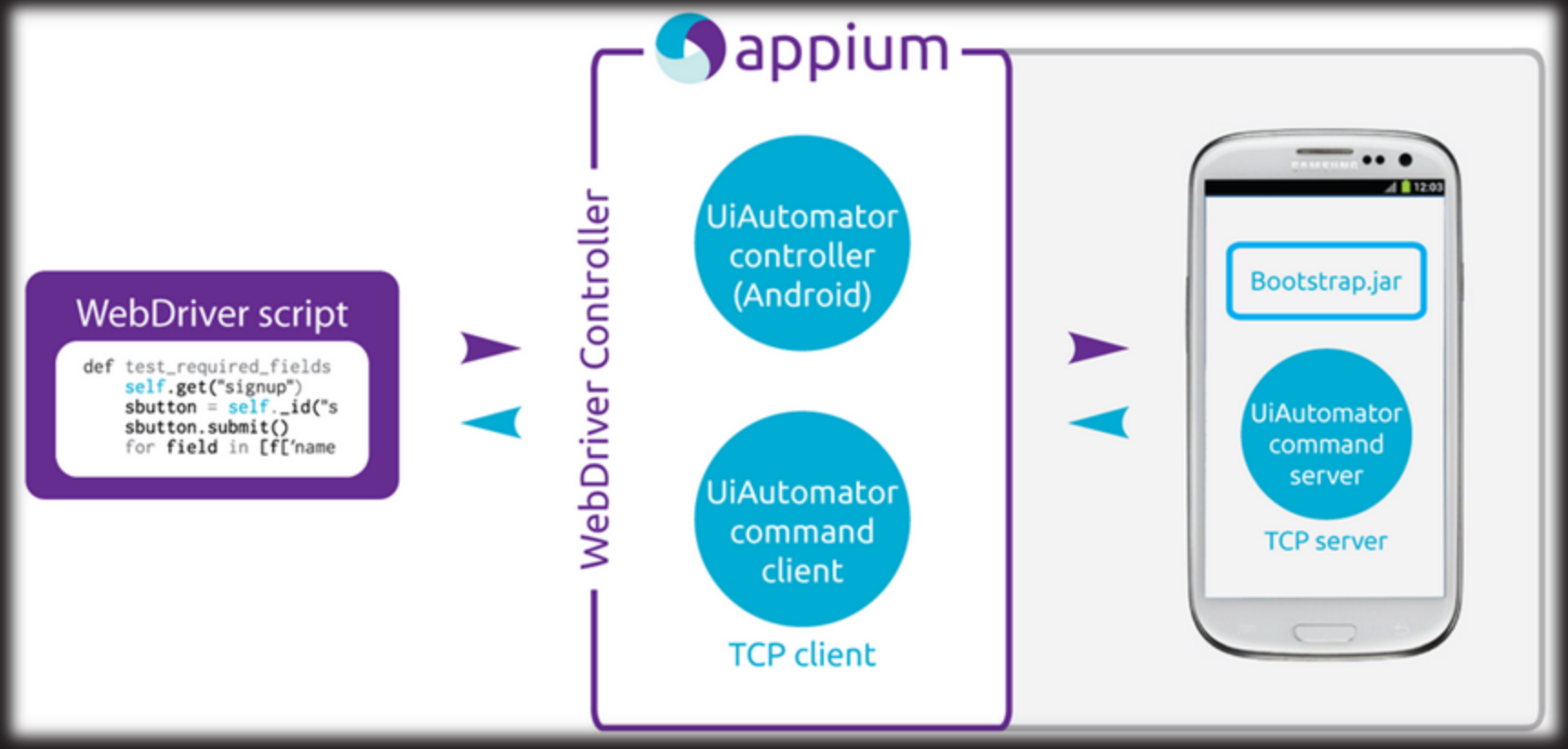
UiAutomator优缺点

- 该框架的几个优点：
  - Google自家推出的，其稳定性和后续的维护更新可以得到保障，运行时也有更多的权限。
  - 可以跨进程操作。
  - 运行速度快。
- 缺点：
  - 不支持Android4.1以下的版本。
  - 不支持Webview，所以一般无法对浏览器应用进行测试。

知识讲解

Appium与UiAutomator关系

- 客户端由脚本 <-> Appium服务端 <-> 手机设备



知识讲解

UiAutomator中的类

- UiAutomator对外提供了UiAutomatorTestCase、UiDevice、UiSelector、UiObject、UiCollection、UiScrollable等重要的类，其各自的作用如下：
  - UiAutomatorTestCase ：

    这个类是继承自Junit TestCase（Junit），对外提供setup、teardown等，以便初始化用例、清除环境等。所以我们在编写的UiAutomator 的脚本时一般都要继承这个类，这样就可以直接使用它的一些方法和Junit单元测试框架中的Assert断言机制。
  - UiObject ：

    UiObject代表一个UI元素，通过UiSelector查找创建一个UiObject实例，找到这个实例以后可以对这个实例进行各类操作。
  - UiDevice ：

    在测试时可以通过getUiDevice() 来实例化UiDevice对象去对设备进行各种控制，如唤醒屏幕，锁屏，点击Home, Back，Menu键等等。

知识讲解

UiAutomator中的类（续1）

Technology Tarena 达内科技

- 其他类的作用如下：
  - UiSelector：

    主要是通过一定查询方式，可以通过UiSelector对象去定位Ui元素。
  - UiCollection：

    UiCollection一般与UiSelector连用，如它的构造函数也要求提供UiSelector参数。

    它的API较少，主要用以从UiSelector筛选出的元素集中挑出所要的元素:getChildByDescription(), getChildByInstance(), getChildByText() ,以及统计元素集的个数getChildCount()。
  - UiScrollable：

    UiScrollable 用来表示可以滑动的界面元素,其继承关系为UiObject -> UiCollection ->UiScrollable。

知识讲解

UiSelector简介

- UiSelector ：专门用于定位Android界面上UI元素的类。
  - 如果发现多个满足条件的控件则会返回第一个控件。
  - 在使用UiSelector的时候可以通过链式调用来组合使用多个属性来定位具体的控件。
  - 还可以使用childSelector()函数来嵌套UiSelector对象。

知识讲解

UiSelector常用API

Technology Tarena 达内科技

- UiSelector类中定位Ui元素
  - 基本定位方法：
    - text
    - description
    - resourceId
    - className
  - 高级定位方法：
    - 组合定位
    - index
    - instance
    - 关系定位（childSelector、fromParent）

知识讲解

Technology Tarena 达内科技

UiAutomator基本定位

UiAutomator定位

- UiAutomator定位是Android自身独有的定位方式，使用UiSelector提供的API来定位Android控件。
- 基本语法：
  - driver.find_element(MobileBy.ANDROID_UIAUTOMATOR, 'xxx')
  - driver.find_element_by_android_uiautomator(xx)
- 例：
  - driver.find_element(MobileBy.ANDROID_UIAUTOMATOR,'new UiSelector().text("7")').click()
  - driver.find_element_by_android_uiautomator('new UiSelector().text("3")').click()

知识讲解

UiSelector定位说明

- 说明：new UiSelector() 是可以省略的。
- 例：
  - driver.find_element(MobileBy.ANDROID_UIAUTOMATOR,'text("7")').click()
    driver.find_element_by_android_uiautomator('text("3")').click()
- 注意：
  - text、resource-id、class、content-desc属性值，实际用driver.find_element_by_android_uiautomator()结合上面的四个方式来直接定位元素，这不太推荐的，因为单单使用那四个方式就完全可以实现定位，一般使用较多的是其高级匹配的定位方式。

知识讲解

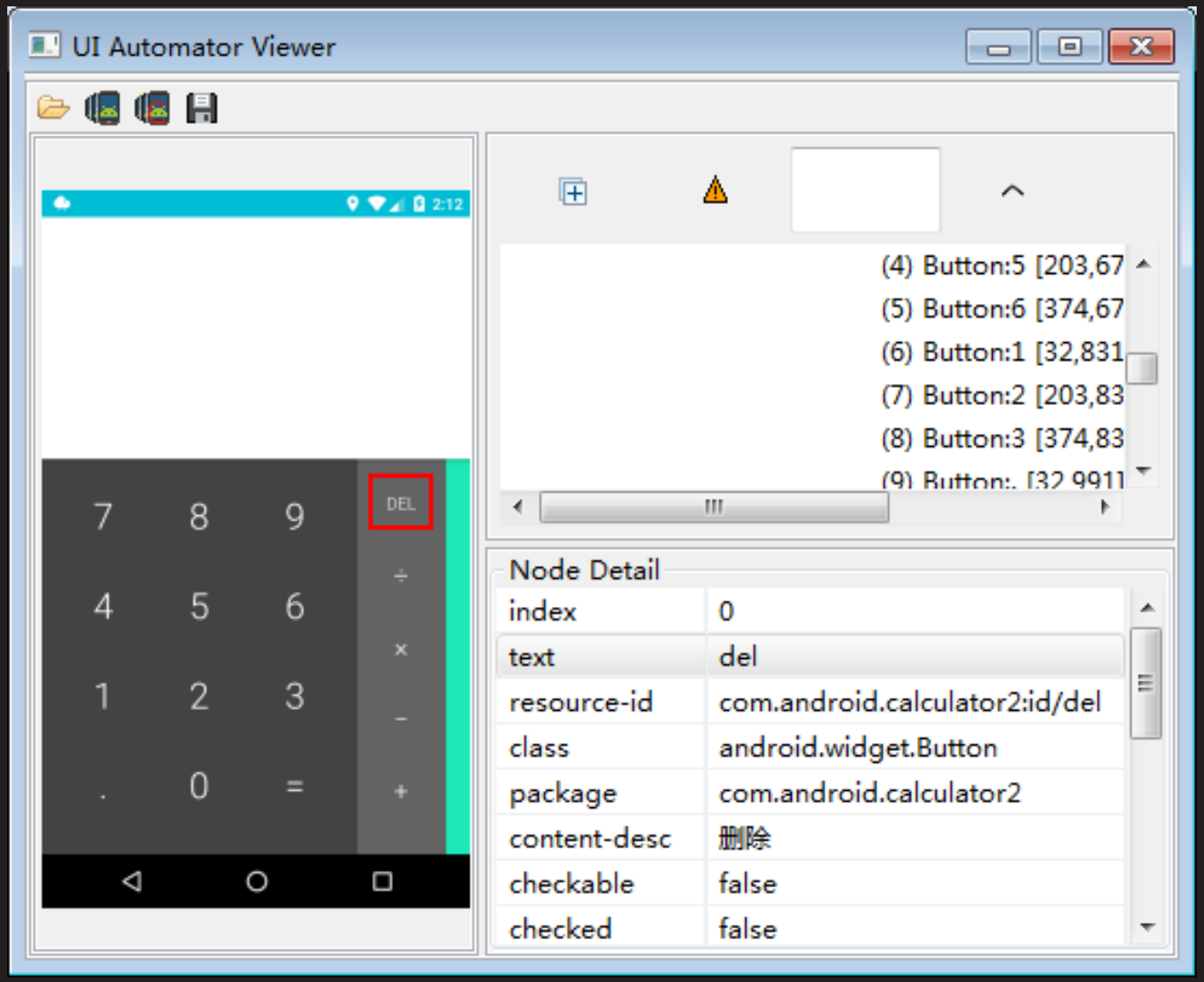
text

- 通过text文本（text属性值）定位，如果文本比较长，可以用textContains或textStartsWith模糊匹配，也可以用正则表达式textMatches匹配。
- 语法：
  - new UiSelector().text("text属性值")
  - new UiSelector().textContains("text属性值")
  - new UiSelector().textStartsWith("text属性值")
  - new UiSelector().textMatches("正则表达式")

知识讲解

text（续1）

- 案例：计算器中的DEL键，text属性值是“del”



知识讲解

text（续2）

- 示例代码：
  - driver.find_element(MobileBy.ANDROID_UIAUTOMATOR,'new UiSelector().text("del")').click()
  - driver.find_element_by_android_uiautomator('new UiSelector().text("del")').click()
  - driver.find_element_by_android_uiautomator('new UiSelector().textContains("e")').click()
  - driver.find_element_by_android_uiautomator('new UiSelector().textStartsWith("de")').click()
  - driver.find_element_by_android_uiautomator('new UiSelector().textMatches(".*e.*")').click()

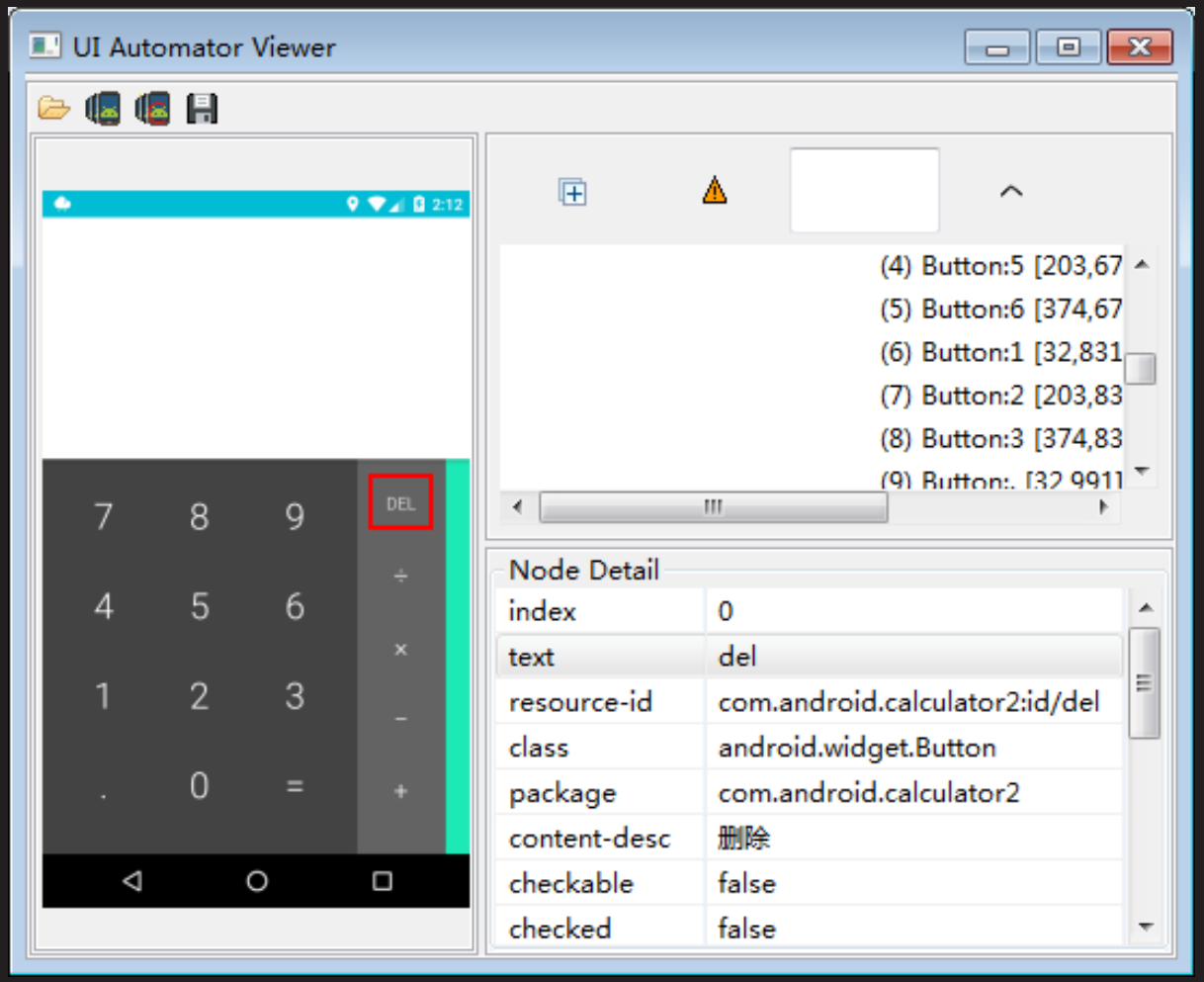
description

- 通过描述（content-desc属性值）定位，如果文本比较长，可以用descriptionContains或descriptionStartsWith模糊匹配，也可以用正则表达式descriptionMatches匹配。
- 语法：
  - new UiSelector().description("content-desc属性值")
  - new UiSelector().descriptionContains("content-desc属性值")
  - new UiSelector().descriptionStartsWith("content-desc属性值")
  - new UiSelector().descriptionMatches("正则表达式 ")

知识讲解

description（续1）

- 案例：计算器中的DEL键，content-desc属性值是“删除”



知识讲解

description（续2）

- 示例代码：
  - driver.find_element(MobileBy.ANDROID_UIAUTOMATOR, 'new UiSelector().description("删除")').click()
  - driver.find_element_by_android_uiautomator('new UiSelector().description("删除")').click()
  - driver.find_element_by_android_uiautomator('new UiSelector().descriptionContains("除")').click()
  - driver.find_element_by_android_uiautomator('new UiSelector().descriptionStartsWith("删")').click()
  - driver.find_element_by_android_uiautomator('new UiSelector().descriptionMatches(".*删.*")').click()

resourceId

Technology Tarena 达内科技

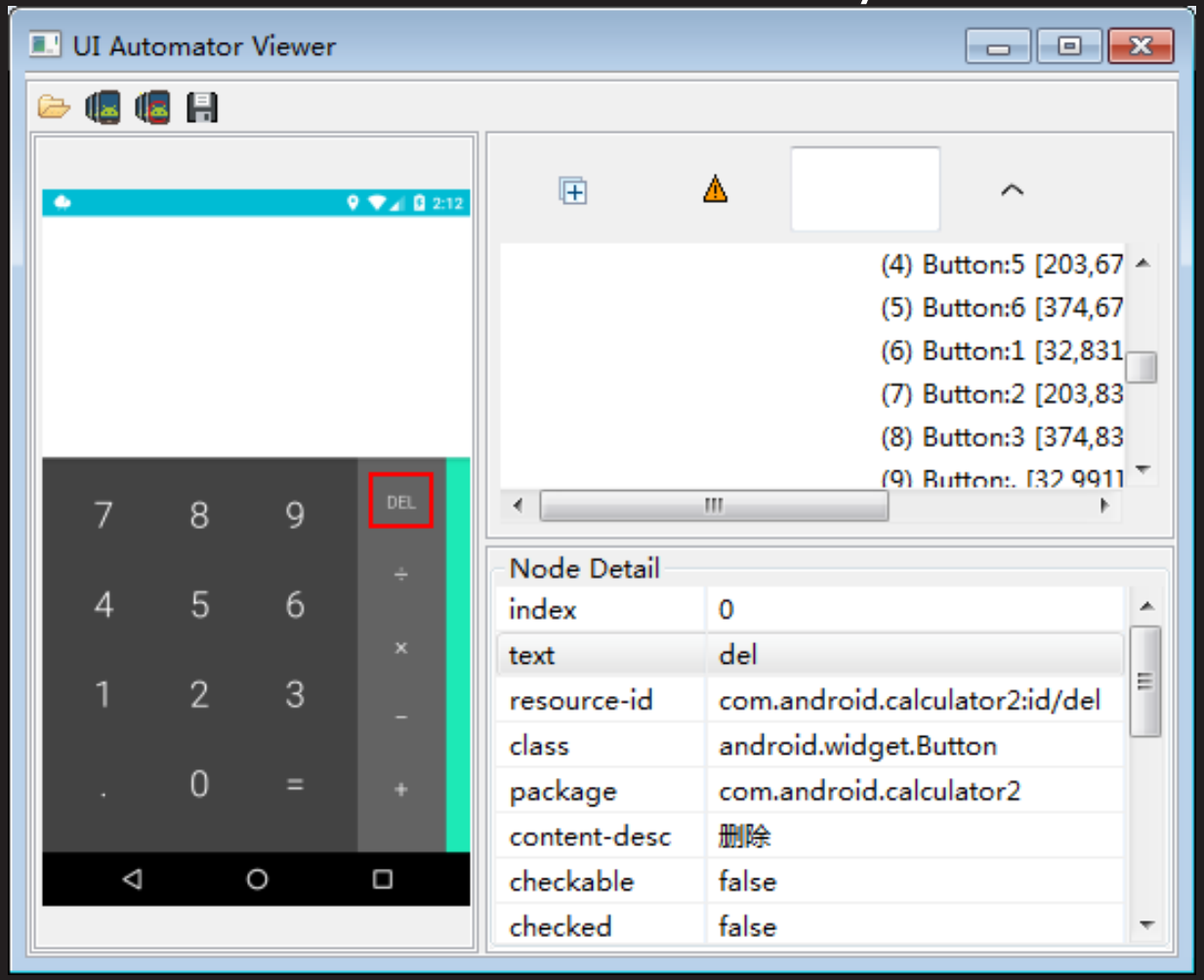
- 通过资源ID（resource-id属性值）定位，如果resourceId比较长，可以用正则表达式resourceIdMatches匹配。
  - new UiSelector().resourceId("resource-id属性值")
  - new UiSelector().resourceIdMatches("正则表达式")
- 说明：与find_element_by_id功能基本一样，但是find_element_by_id可以将id属性值简写，去掉前面的”包名:id”部分，但find_element_by_android_uiautomator不能使用简写的id属性值。

知识讲解

注意：
没有resourceIdContains方法和resourceIdStartsWith方法。
android4.3及以上系统才可通过资源ID的方式定位控件。

resourceId（续1）

- 案例：计算器中的DEL键，resource-id属性值是“com.android.calculator2:id/del”



知识讲解

resourceId（续2）

- 示例代码：
  - driver.find_element(MobileBy.ANDROID_UIAUTOMATOR,'new UiSelector().resourceId("com.android.calculator2:id/del")').click()
  - driver.find_element_by_android_uiautomator('new UiSelector().resourceId("com.android.calculator2:id/del")').click()
  - driver.find_element_by_android_uiautomator('new UiSelector().resourceIdMatches(".*e.*")').click()

知识讲解

resourceId（续3）

Technology Tarena 达内科技

- 注意： 在UI Selector中如果resource-id属性值是“com.android.calculator2:id/del” ，不能简写为del。
  - 正确写法：

```
driver.find_element_by_id("com.android.calculator2:id/del").click()
driver.find_element_by_id("del").click()
```

  - 正确写法：

```
driver.find_element_by_android_uiautomator('new UiSelector().resourceId("com.android.calculator2:id/del")').click()
```

  - 错误写法：

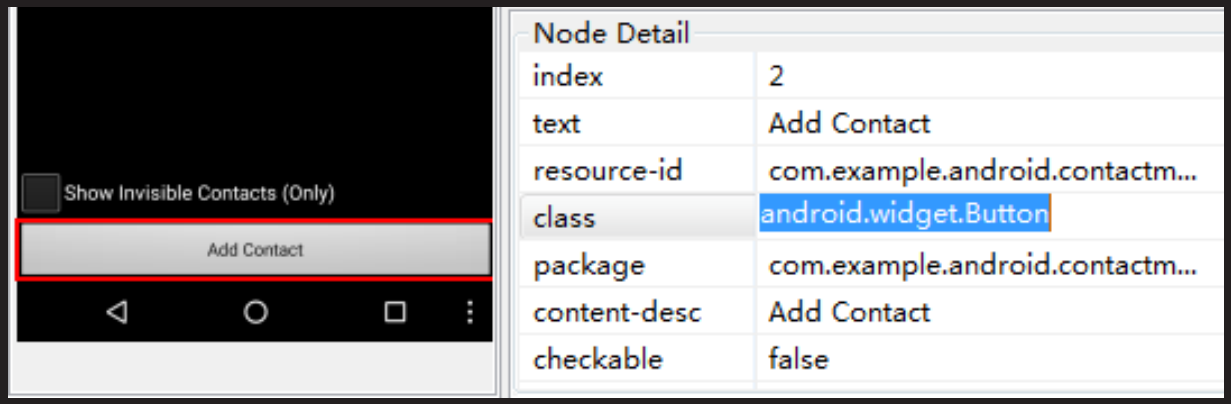
```
driver.find_element_by_android_uiautomator('new UiSelector().resourceId("del")').click()
```

知识讲解

className

- 通过class属性值来定位，页面上的class属性一般不唯一，一般用于定位多个元素，用下标（index或instance）获得指定编号的那一个来操作。
  - new UiSelector().className("class属性值")
  - new UiSelector().classNameMatches("正则表达式")
- 例：
  - new UiSelector().className("android.widget.Button")
  - new UiSelector().classNameMatches(".*Button")

知识讲解

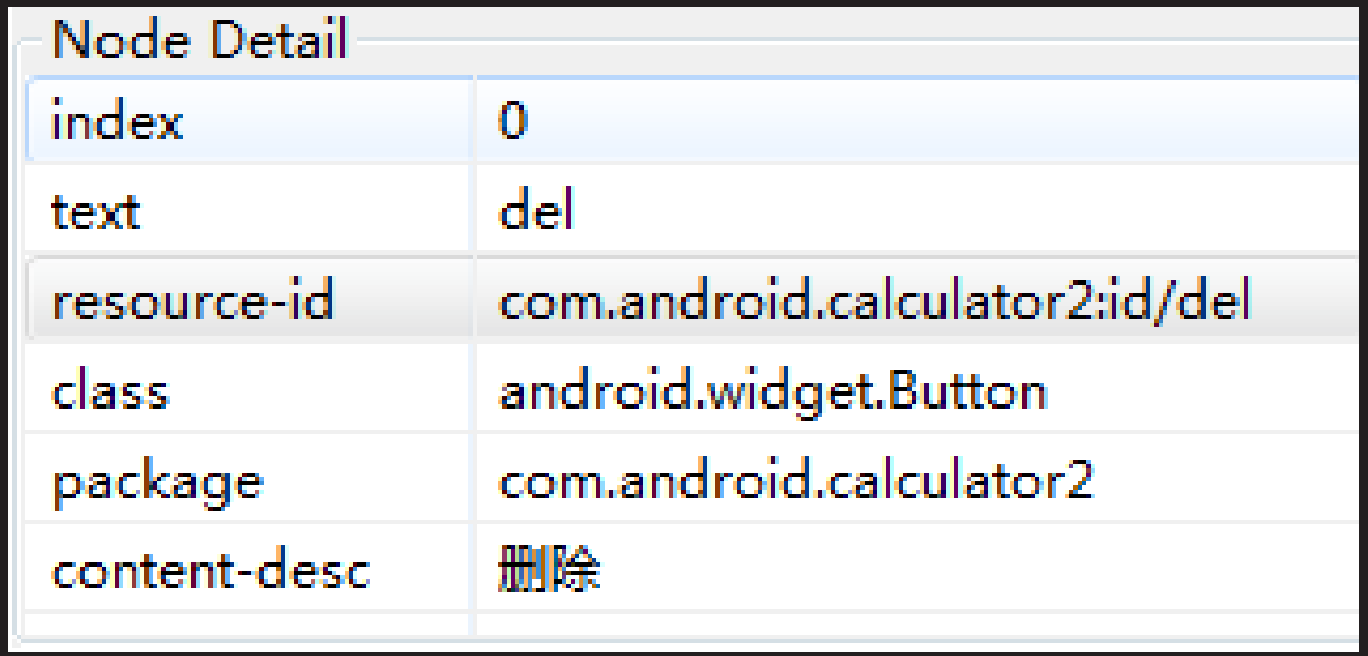


Technology Tarena 达内科技

UiAutomator高级定位

组合定位

- id与text属性组合
- 例：
  - driver.find_element_by_android_uiautomator('new UiSelector().resourceId("com.android.calculator2:id/del").text("del")').click()



知识讲解

组合定位（续1）

Technology Tarena 达内科技

- class与text属性组合
- 例：
  - driver.find_element_by_android_uiautomator('new UiSelector().className("android.widget.Button").text("del")').click()

知识讲解

Node Detail

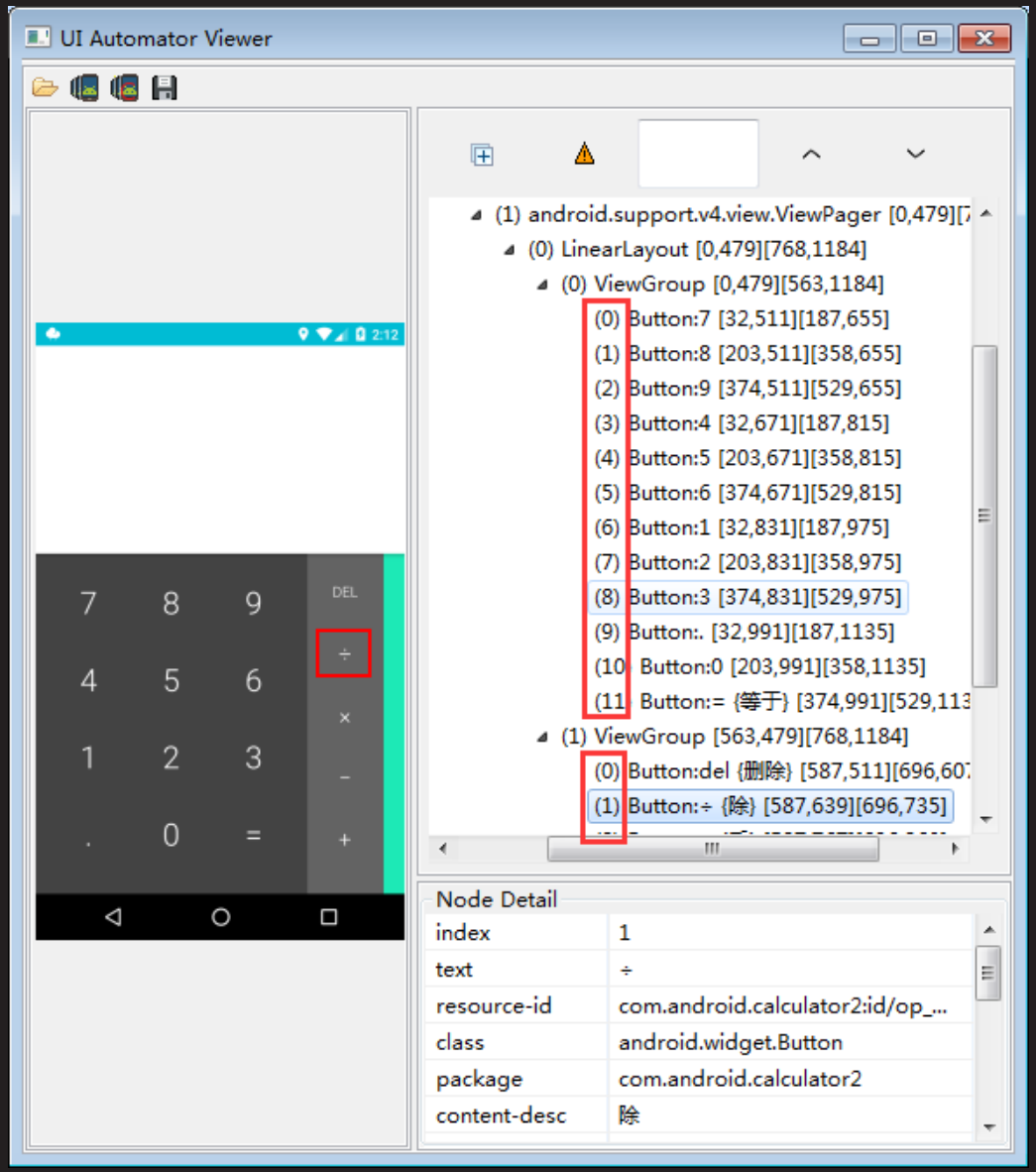
index	0
text	del
resource-id	com.android.calculator2:id/del
class	android.widget.Button
package	com.android.calculator2
content-desc	删除

index

- index：在同一级中的编号，在兄弟类中组件的编号，index从0开始。
- 例：
  - driver.find_element(MobileBy.ANDROID_UIAUTOMATOR,'new UiSelector().className("android.widget.Button").index(2)').click()
  - driver.find_element_by_android_uiautomator('new UiSelector().className("android.widget.Button").index(3)').click()

知识讲解

index（续1）



说明：index是在同一级中的编号。

例如：无法定位到“删除”按钮

```
driver.find_element_by_android_uiautomator('new UiSelector().className("android.widget.Button").index(12)').click()
```

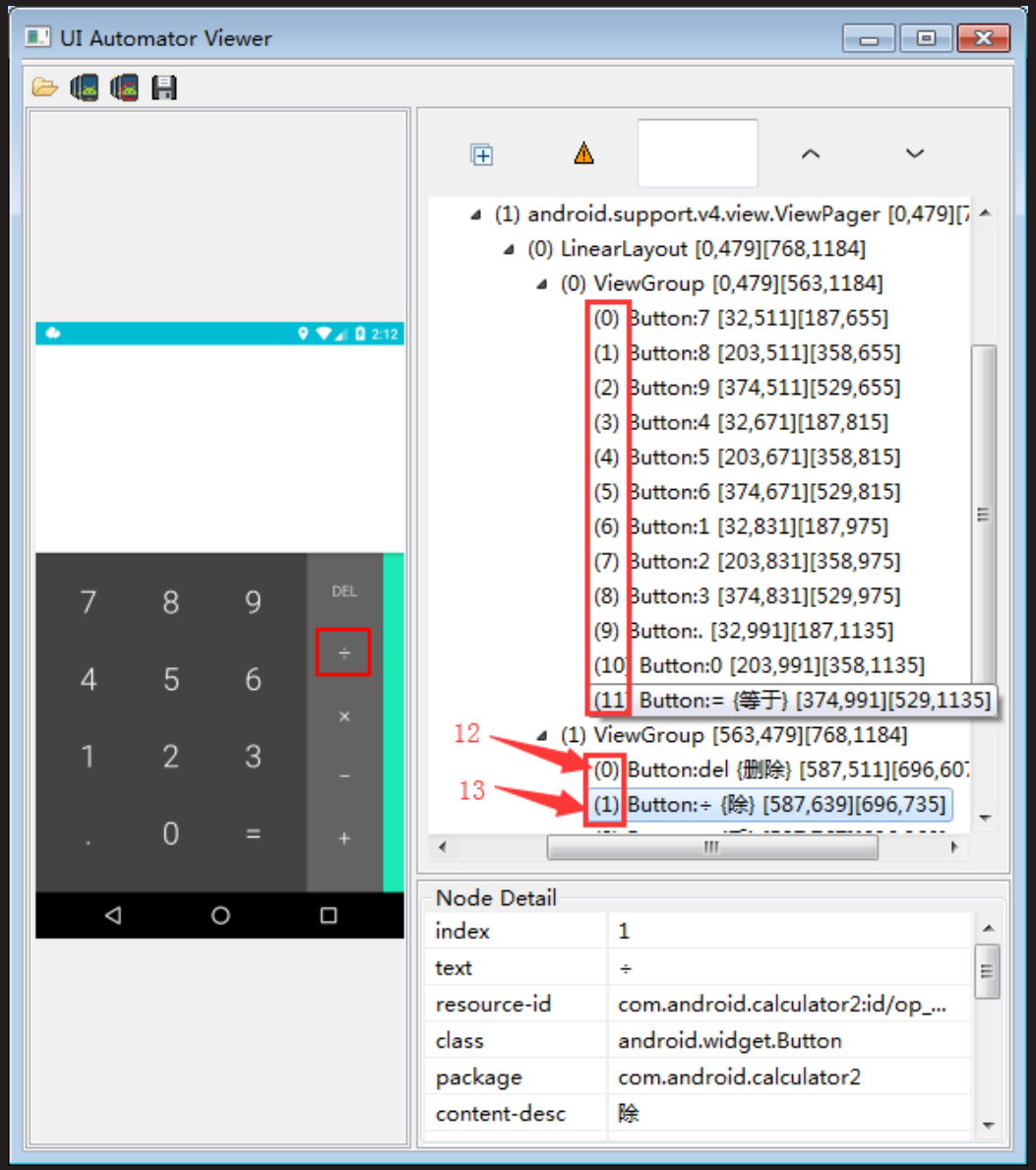
知识讲解

instance

- instance：同一个布局中同一类组件的编号，instance从0开始。
- 例：
  - # del
  - driver.find_element(MobileBy.ANDROID_UIAUTOMATOR,'new UiSelector().className("android.widget.Button").instance(12)').click()
  - driver.find_element_by_android_uiautomator('new UiSelector().className("android.widget.Button").instance(12)').click()

知识讲解

instance（续1）

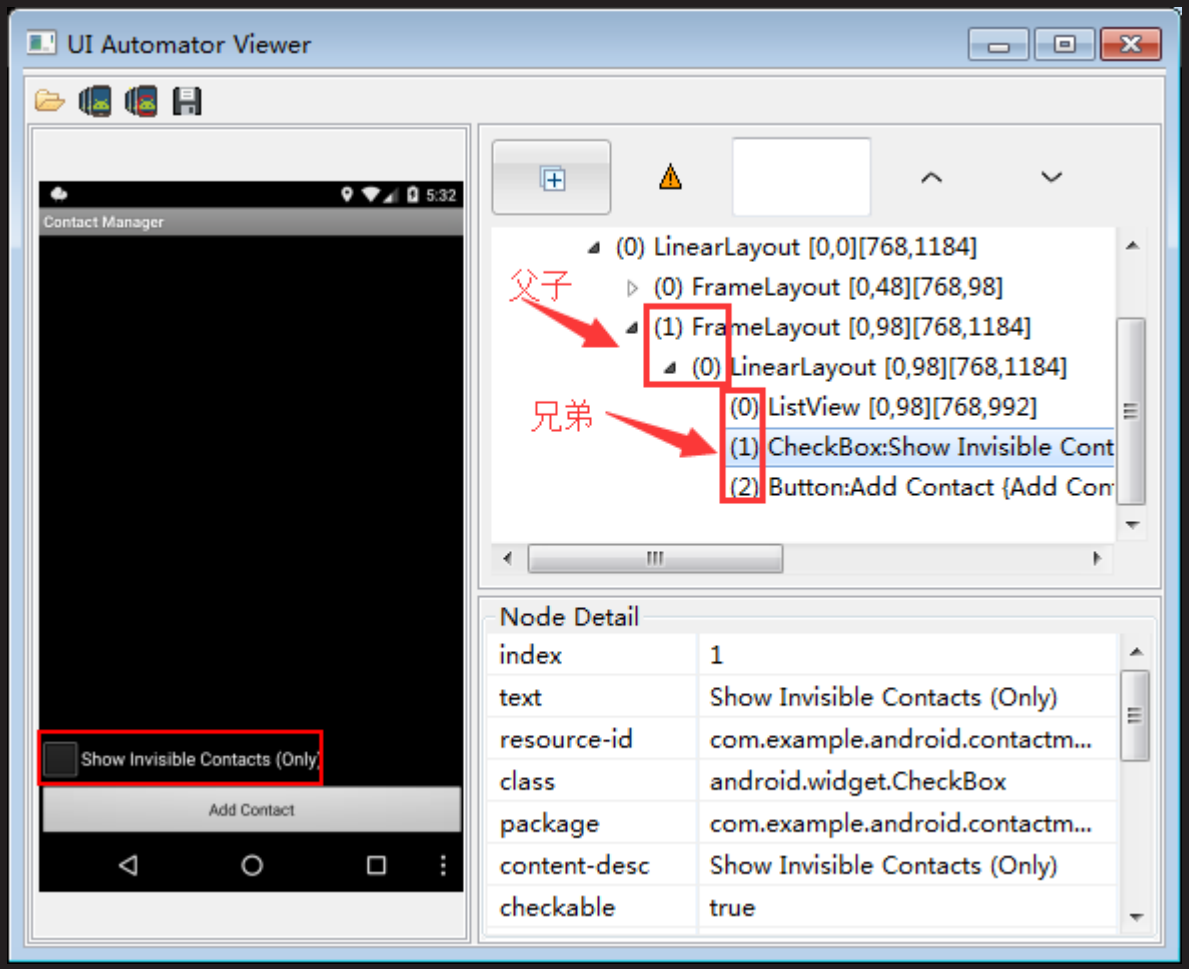


知识讲解

说明：instance是同一个布局中同一类组件的编号，
例如：能定位到“删除”按钮
driver.find_element_by_android_uiautomator('new UiSelector().className("android.widget.Button").instance(12)').click()

关系定位

- 父子定位childSelector：通过父元素找到子元素。
- 兄弟定位fromParent：找到同一父级元素下的兄弟元素。



知识讲解

childSelector

- 父子定位childSelector：如果不能直接定位某个目标元素，但是它的父元素很好定位，就先定位父元素，通过父元素查找到目标子元素。
- 例：
  - driver.find_element_by_android_uiautomator('resourceId("android:id/content").childSelector(className("android.widget.LinearLayout").childSelector(text("Show Invisible Contacts (Only)")))').click()
  - driver.find_element_by_android_uiautomator('new UiSelector().className("android.widget.LinearLayout").childSelector(new UiSelector().text("Show Invisible Contacts (Only)"))').click()

知识讲解

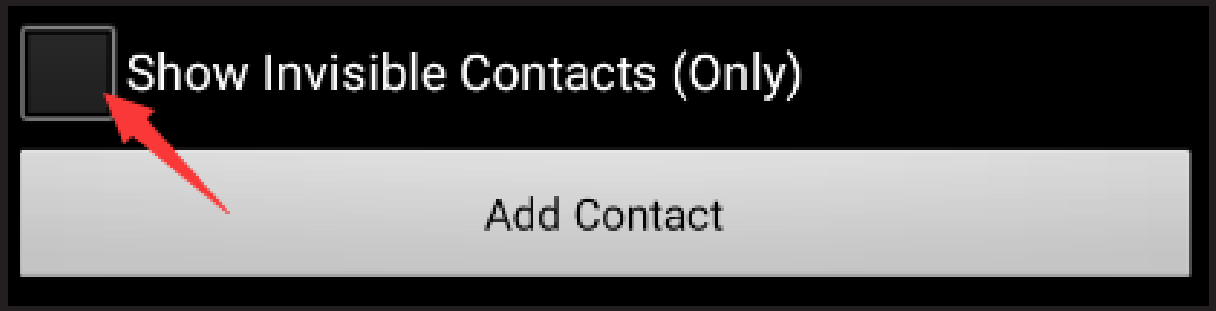
fromParent

- 兄弟定位fromParent：如果目标元素和其父元素都不好定位，但是跟他相邻的兄弟元素很好定位，就可以通过兄弟元素，找到同一父级元素下的目标元素。
- 例：
  - driver.find_element_by_android_uiautomator('text("Show Invisible Contacts (Only)").fromParent(className("android.widget.Button"))').click()

知识讲解

通讯录关系定位

- 使用UiAutomator定位ContactManager中通过Add Contact按钮查找到上面的复选框。
- 详见【COOKBOOK】



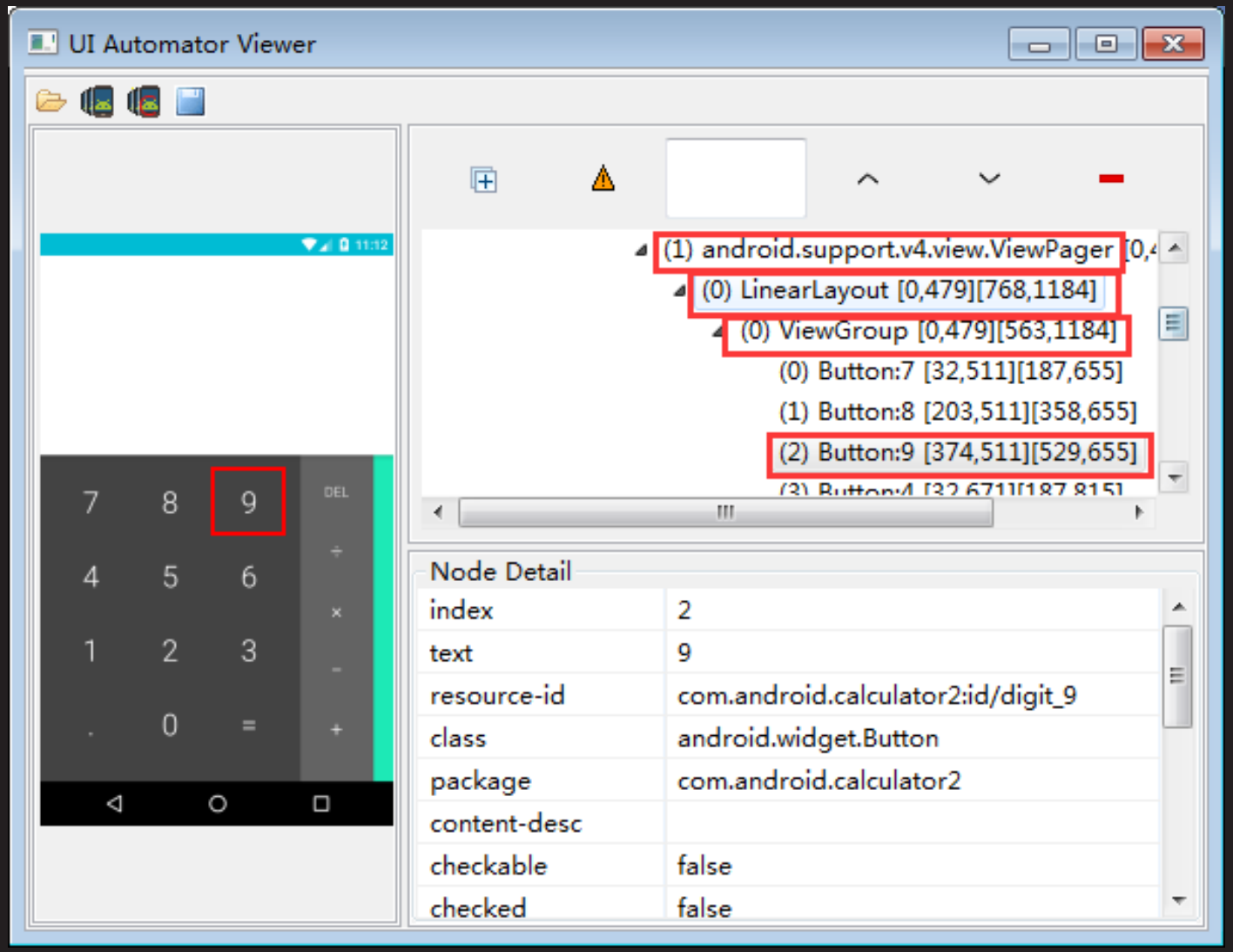
课堂练习

Technology Tarena 达内科技

其他定位方法

层级定位

- 先定位到某个祖先节点，再定位其后代节点



知识讲解

层级定位（续1）

- 例1：逐级查找
  - driver.find_element(MobileBy.XPATH,"//android.support.v4.view.ViewPager")
    .find_element(MobileBy.XPATH,"//android.widget.LinearLayout")
    .find_element(MobileBy.XPATH,"//android.view.ViewGroup[1]")
    .find_element(MobileBy.XPATH,"//android.widget.Button[3]").click()

第一层定位

第二层定位

第三层定位

第四层定位

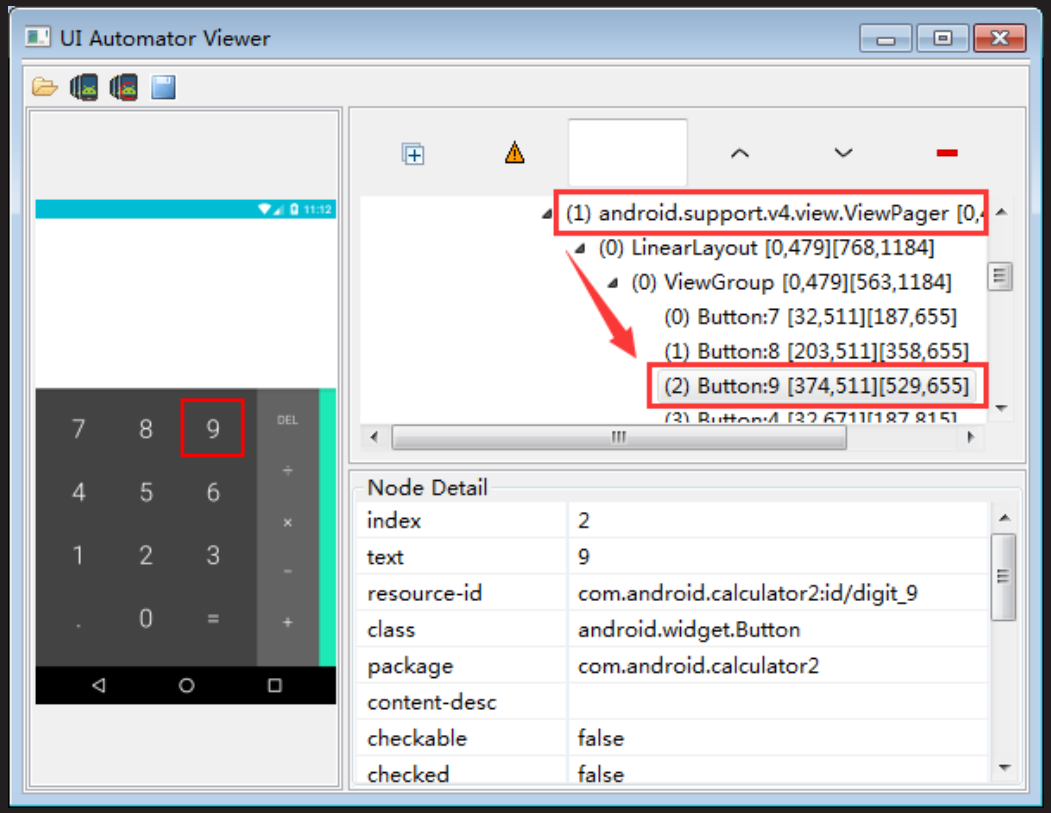
知识讲解

层级定位（续2）

- 例2：跳级查找
  - driver.find_element(MobileBy.XPATH,"//android.support.v4.view.ViewPager").find_element(MobileBy.XPATH,"//android.widget.Button[3]").click()

第一层定位

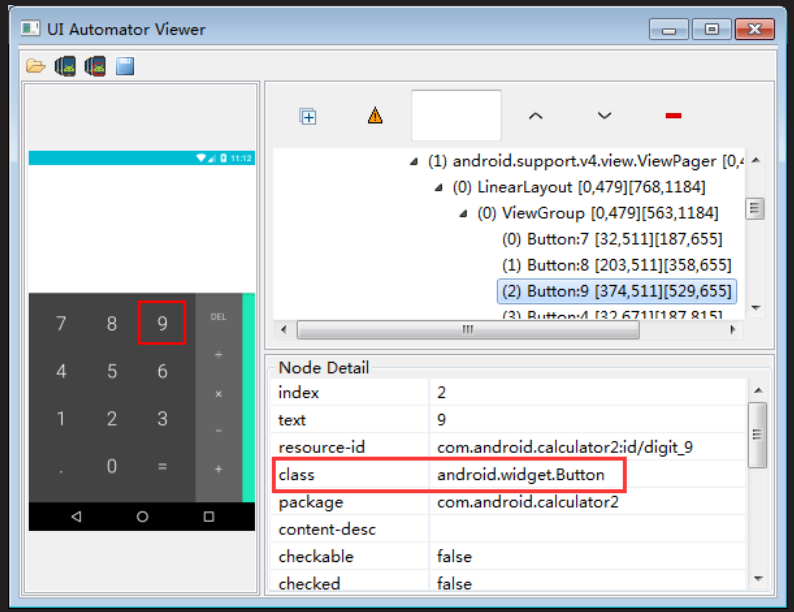
第二层定位



知识讲解

按编号定位

- 通过findElements或findElementsByXXX查找到多个页面元素后，使用编号(索引号、顺序号)来定位其中指定的一个。
- 例1：按编号定位和ClassName定位一起使用。
  - driver.find_elements(MobileBy.CLASS_NAME,"android.widget.Button")[2].click()

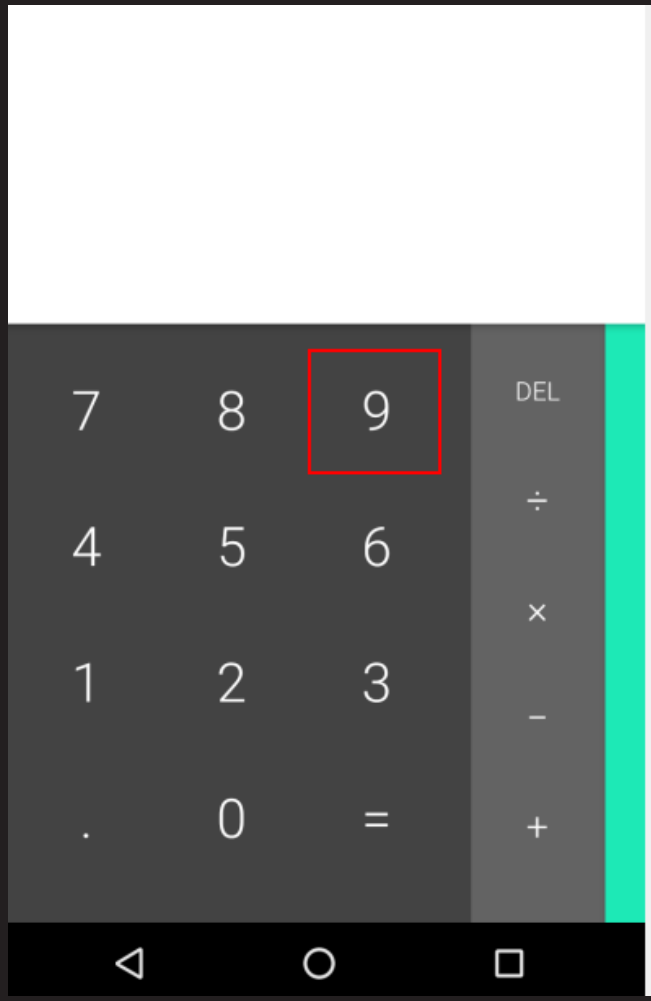


知识讲解

按编号定位（续1）

Technology Tarena 达内科技

知识讲解




```
(0) FrameLayout [0,0][768,1184]
  (0) LinearLayout [0,0][768,1184]
    (0) FrameLayout [0,48][768,1184]
      (0) LinearLayout [0,48][768,1184]
        (0) RelativeLayout [0,48][768,479]
        (1) android.support.v4.view.ViewPager [0,479][768,1184]
          (0) LinearLayout [0,479][768,1184]
            (0) ViewGroup [0,479][563,1184]
              (0) Button:7 [32,511][187,655]
              (1) Button:8 [203,511][358,655]
              (2) Button:9 [374,511][529,655]
              (3) Button:4 [32,671][187,815]
              (4) Button:5 [203,671][358,815]
              (5) Button:6 [374,671][529,815]
              (6) Button:1 [32,831][187,975]
              (7) Button:2 [203,831][358,975]
              (8) Button:3 [374,831][529,975]
              (9) Button:. [32,991][187,1135]
              (10) Button:0 [203,991][358,1135]
              (11) Button:= {等于} [374,991][529,1135]
```

Node Detail	
index	2
text	9
resource-id	com.android.calculator2:id/digit_9
class	android.widget.Button
package	com.android.calculator2
content-desc	
checkable	false
checked	false

按编号定位（续2）

- 例2：按编号定位和xpath定位一起使用
  - driver.find_elements_by_xpath("//android.support.v4.view.ViewPager/android.widget.LinearLayout/android.view.ViewGroup[1]/android.widget.Button")[2].click()
- 例3：按编号定位和层级定位一起使用
  - driver.find_elements(MobileBy.XPATH,"//android.support.v4.view.ViewPager/android.widget.LinearLayout/android.view.ViewGroup")[0].find_elements(MobileBy.XPATH,"//android.widget.Button")[2].click()

第一层定位

第二层定位

知识讲解

按编号定位计算器

- 按编号定位计算器实现3+5=的步骤
- 详见【COOKBOOK】

课堂练习

Technology Tarena 达内科技

总结和答疑