

如何使用 monkeyrunner 与 view 进行交互

android 官方里的 monkeyrunner 只介绍了一些有限的 ui 测试方法. 比如只能指定坐标点模拟操作行为, 测试结果对比只能通过截图进行简单对比结果. 当然我们可以聪明的来实现猜测到 **view** 视图的坐标点, 那么我们就可以像 robotium 一样, 在 monkeyrunner 脚本中, 对控件级的 **view** 视图进行交互操作.

如何实现捕捉到当前 **activity** 展现的所有的控件信息呢? monkeyrunner.jar 里目前有 EasyMonkeyDevice.class 和 By.class, 提供了一些新的 api, 很大的满足了一些高级需求.

有几个 blog 文章对 EasyMonkeyDevice 类的 api 进行了一些示例, 非常浅显易懂.

yan1391 的博客 [Android 测试之初试 Monkeyrunner](#). 国外的 diego 也写了一系列关于 [monkeyrunner 的文章](#), 并自己写了一些 [viewclient apis](#) 可以下载到, 非常不错. 请大家参考, 但是 diego 把博客写到了 blogspot 上, 哈, 所以大陆的朋友们就郁闷了. 只能翻墙看了.

其实以上所有文章主要通过使用了 By, EasyMonkeyDevice, monkeydevice 这 3 个类来实现与 view object 的交互. 那我们就对这 3 个类进行简单的剖析一下.

访问[这里](#) 可以看到 [EasyMonkeyDevice](#) 和 [By](#) 的源码 (在线浏览).

1. By.class 可以根据 id 获得 viewobject. 例如我们有一个 text view 的 id 是 id/account 那么我们就可以通过如下获得该 view object.

```
account = By.id('id/account')
```

2. EasyMonkeyDevice.class 获得 view object 后, 可以用 EasyMonkeyDevice 类进行一些操作. getText(), touch(), type(). 分别为获得该 view object 的文本值, 进行触摸屏操作 (其会获取 view object 的坐标值, 并进行操作). type() 则会先 touch, 然后输入字符.

3. monkeyDevice [在线浏览源码](#)

```
# 通过 MonkeyDevice 获取 HierarchyViewer 对象实例
```

```
hierarchy_viewer = device.getHierarchyViewer()
```

```
#通过 HierarchyViewer 获取 ViewNode 实例 (一个 viewnode 实例代表一个控件)
```

```
view_node = hierarchy_viewer.findView(By.id('id/emptyText'))
```

```
#获取控件内的文本
```

```
text = view_node.namedProperties.get('mText').toString()
```

```
if text.find('have any') < 0:
```

```
print 'not find have any!'
```

```
else:
```

```
print 'find have any!'
```

```
print text
```

总结： 如此这般，我们用 monkeyrunner 也可以写出和 robotium 一样的脚本，可以对 view object 进行操作。 而且可以和坐标点值的方法进行融合。 可以由很大的突破。

另发现了一些问题，目前怀疑是 android monkeyrunner.jar 的 bug。问题是 EasyMonkeyDevice 的 touch() 函数，不能对应用程序弹出的对话框正确的定位坐标值。 当一个应用程序弹一个对话框时（确认还是取消之类的对话框），当根据 id touch 确定和取消时， 定位的坐标点错误，导致无法实现想要的功能。 还未具体分析时什么时候对话框会产生能该问题。 为什么说可能是 bug, 是因为用 tools 里的 HierarchyViewer 进行察看时，发现该对话框的实际坐标显示位置有问题。 但渲染到手机屏幕确是正常的。 这导致， 通过 hierachy 获得的坐标值有问题，点击操作无法 触发对应的 button view。

Monkeyrunner API 帮助文档

(2013-04-11 17:01:05)

转 载 ▼

标签： 分类： [Monkeyrunner](#)

it

Monkeyrunner API 帮助文档

文档下载地址 <http://vdisk.weibo.com/s/xcHlw/1365674074>

内容为 MonkeyRunner 帮助

1、com.android.monkeyrunner.MonkeyRunner.alert

当前脚本的运行过程中显示警告对话框。对话框是模态，因此脚本停止，直到用户关闭对话框。

参数：

消息 - 消息显示在对话框。

标题 - 对话框的标题。默认值是“提醒”。

okTitle - 使用对话框中的按钮的文本。默认值是“OK”。

返回：返回任何内容。

2、`com.android.monkeyrunner.MonkeyDevice.broadcastIntent` 发送一个广播设备的意图。

参数：

URI - 意图的 URI。 行动 - 该行动的意图。 数据 - 数据的 URI 意图

的 `mimetype` - mime 类型的意图。 类别 - 迭代的意图类别名称。

演员 - 一个额外的字典添加到意图。这些额外的类型推断的值的 Python 类型。

组件 - 组件的意图。

标志 - 一个迭代的标志为 `Intent.All` 参数是可选的。

每个参数的默认值是 `null`。（见 `android.content.Context.sendBroadcast`（意向））

返回：返回任何内容。

3、`com.android.monkeyrunner.MonkeyRunner.choice`

显示一个选择对话框，允许用户选择从一个单一的项目，一个项目清单。

参数：

消息 - 提示信息显示在对话框。

选择 - 一个可迭代的 Python 类型的选择列表显示

标题 - 对话框的标题。默认的是“输入”

返回：所选项目的抵消了 `Iterable` 基于 0 的数字。

4、`com.android.monkeyrunner.MonkeyImage.convertToBytes`

转换成特定格式的 `MonkeyImage` 作为一个字符串返回结果。使用此获得 `rawpixels` 在一个特定的格式。字符串输出是有更好的表现。

参数：

格式 - 目标格式（例如，“PNG 便携式网络图形格式”。默认为 PNG。

返回：一个 `String` 生成的图像。

5、`com.android.monkeyrunner.MonkeyDevice.drag` 模拟设备屏幕上拖动（触摸，持有和移动）。

参数：

开始 - 拖动起点（以像素为单位元组 `(X, Y)`）

结束 - 阻力终点（以像素为单位元组 `(X, Y)`）

时间 - 在几秒钟内拖时间（默认为 1.0 秒）

步骤 - 插值点时要采取的步骤。（默认是 10）

返回：返回任何内容。

6、`com.android.monkeyrunner.MonkeyView.getAccessibilityIds` 返回当前视图的辅助 IDS

返回：视图的辅助 IDS 作为一个整数列表

7、`com.android.monkeyrunner.MonkeyRect.getCenter`

返回一个两个项目清单，其中包含矩形的中心的 `x` 和 `y` 的值

返回：该中心作为一个整数的两个项目清单坐标

8、`com.android.monkeyrunner.MonkeyView.isChecked` 获取视图的选中状态

返回：一个布尔值，该项目是否被选中或不

9、`com.android.monkeyrunner.MonkeyView.getChildren` 返回当前视图的孩子们

返回：认为儿童 `MonkeyView` 对象名单

10、`com.android.monkeyrunner.MonkeyView.isEnabled` 返回启用状态的看法。

返回：作为一个布尔启用状态

- 11、com.android.monkeyrunner.MonkeyView.getFocused 视图返回集中状态
返回：集中的状态作为一个布尔
- 12、com.android.monkeyrunner.MonkeyRect.getHeight 返回矩形的高度。
返回：作为一个整数的矩形的高度
- 13、com.android.monkeyrunner.MonkeyDevice.getHierarchyViewer 获取设备 HierarchyViewer 对象。
返回一个 HierarchyViewer 对象
- 14、com.android.monkeyrunner.MonkeyView.getLocation 返回视图的位置在一个 MonkeyRect 形式
返回：视图作为 MonkeyRect 对象的位置
- 15、com.android.monkeyrunner.MonkeyView.getParent 返回当前视图的父
返回：视图的父作为 MonkeyView 对象
- 16、com.android.monkeyrunner.MonkeyDevice.getProperty 由于设备上的变量的名称，返回变量的值
参数：
关键 - 变量的名称。可用的名称在
<http://developer.android.com/guide/topics/testing/monkeyrunner.html> 上市。
返回：该变量的值
- 17、com.android.monkeyrunner.MonkeyDevice.getPropertyList 检索，可以查询的属性
返回：返回任何内容。
- 18、com.android.monkeyrunner.MonkeyImage.getRawPixel
一个单一的 ARGB (alpha, 红色, 绿色, 蓝色) 像素位置的 X, Y。参数 X 和 Y 是 0, 表示在像素尺寸。X 增加的权利, 对底部的 Y 增加。此方法返回一个 tuple。
参数：
X - x 偏移的像素 Y - 像素的 y 偏移量
返回的元组 (A, R, G, B) 的像素。tuple 中的每个项目都有 0-255 范围内。
- 19、com.android.monkeyrunner.MonkeyImage.getRawPixelInt
一个单一的 ARGB (alpha, 红色, 绿色, 蓝色) 像素位置的 X, Y。参数 X 和 Y 是 0, 表示在像素尺寸。X 增加的权利, 对底部的 Y 增加。此方法返回一个整数。
参数：
X - x 偏移的像素 Y - 像素的 y 偏移量
返回：X, Y 的无符号整数像素。高阶 8 位是 A, followedby 8 位为 R, G 和 8 8 为 B
- 20、com.android.monkeyrunner.MonkeyDevice.getRootView 获取当前根视图
返回：根视图对象
- 21、com.android.monkeyrunner.MonkeyView.getSelected 返回视图的选中状态
返回：作为一个布尔选中状态
- 22、com.android.monkeyrunner.MonkeyImage.getSubImage 复制图像的矩形区域。
参数：
矩形 - 一个元组 (X, Y, W, H), 说明该地区复制。x 和 y 指定上左边角落的区域。w 是该地区的像素宽度, h 是它的高度。 返回：MonkeyImage 对象代表复制的区域。
- 23、com.android.monkeyrunner.MonkeyDevice.getSystemProperty 代名词的 getProperty ()
参数：
关键 - 系统变量的名称。 返回：该变量的值。

24、`com.android.monkeyrunner.MonkeyView.getText` 返回视图所载文本

返回：在视图中的文本

25、`com.android.monkeyrunner.MonkeyDevice.getViewByAccessibilityIds` 获得指定的无障碍标识的看法。

参数：

`windowId` - 视图来检索窗口的 ID。 无障碍标识 - 辅助视图来检索标识。 返回：具有指定 ID 的视图对象。

26、`com.android.monkeyrunner.MonkeyDevice.getViewById` 取得具有指定 id 的观点。

参数：

ID - ID 的视图来检索。

返回：具有指定 ID 的视图对象。

27、`com.android.monkeyrunner.MonkeyView.getViewClass` 返回视图类的名称。