

# rMonkeyRunner

这是 `realMonkey` 最核心的一个类，一个 `rMonkeyRunner` 类代表一台设备或者模拟器。这个类被用来控件你的设备或者模拟器。里面包括 UI 事件，查看当前页面信息，安装，卸载或者启动你的应用程序，等等其他更复杂的操作。

Methods	
void	<code>assertActivity(String expected, Boolean result)</code> 验证预期的 Activity 值是否与设备当前屏幕上的 Activity 相等
void	<code>assertEqual(String actual, String expected, Boolean result)</code> 验证实际值与预期的值是否相等
void	<code>assertExist(String type, String value, Int match, Boolean scrollable)</code> 验证设备屏幕上存在指定的控件
void	<code>assertText(String expectText, String type, String value, Int match)</code> 验证指定条件的控件 text 属性值与预期相等
void	<code>clearAppData(String packageName)</code> 清除指定安装包的缓存数据
void	<code>clearTextField(String type, String value, Int match)</code> 清空指定条件的文本框内容
void	<code>click( String type, String value, Int match)</code> 通过指定条件在指定控件上发送一个点击事件
void	<code>clickAndWaitForNewWindow(String type, String value, Int match, Int timeout)</code> 在 timeout 时间内找到指定条件的控件，并在相应控件位置发送一个点击事件
void	<code>clickxy(String x, String y)</code> 发送一个指定 x,y 坐标的点击事件到设备屏幕上
void	<code>closeApp(String packageName)</code> 关闭设备上包名为 packageName 的应用程序
void	<code>drag(String direction)</code> 在设备屏幕上模拟一个滑动手势
void	<code>dragxy(Int x, Int y, Int toX, Int toY)</code> 通过指定起点坐标和终点坐标，在设备屏幕上模拟一个滑动手势
void	<code>dragToEnd(String type, String value, Int match, int steps)</code> 向下滚动指定控件，直到控件无法向下滚动或者滚动次数达到 steps 值为止
void	<code>dragToBegin(String type, String value, Int match, int steps )</code> 向上滚动指定控件，直到控件无法向上滚动或者滚动次数达到 steps 值为止
UiElement[]	<code>getAllElements()</code> 获取设备当前屏幕上所有控件的集合
String	<code>getClassName(String type, String value, Int match)</code> 通过指定条件的控件返回控件的 class 属性值
String	<code>getCurrentActivityName()</code> 获取设备当前屏幕的 Activity 值
UiElement	<code>getElement(String type, String value, Int match)</code> 返回指定条件的控件

Method	
UiElement[]	<b>getElements</b> ( <i>String type, String value</i> ) 返回设备当前屏幕上满足指定条件的所有控件集合
String	<b>getSystemProp</b> ( <i>String key</i> ) 获取设备的相关属性，等同于 adb shell getprop <key>
String	<b>getText</b> ( <i>String type, String value, Int match</i> ) 通过指定条件的控件返回控件的 text 属性值
(String,String)	<b>getXY</b> ( <i>String type, String value, Int match</i> ) 通过指定条件的控件返回控件在设备屏幕上的坐标
void	<b>input</b> ( <i>String text, String type, String value, Int match</i> ) 在指定条件的控件上输入 text
void	<b>install</b> ( <i>String apkPath</i> ) 安装 apkPath 下的应用到设备中
void	<b>longclick</b> ( <i>String type, String value, Int match</i> ) 在指定条件的控件上发送一个长按事件
void	<b>press</b> ( <i>String name</i> ) 发送一个 keycode 事件到设备中
Popen	<b>shell</b> ( <i>String cmd</i> ) 执行一条终端命令
void	<b>sleep</b> ( <i>Float seconds</i> ) 暂停执行脚本 seconds 时间
void	<b>startActivity</b> ( <i>String component</i> ) 启动设备上指定条件的应用程序
void	<b>swipe</b> ( <i>String direction,String type,String className,Int match,Int speed</i> ) 将设备屏幕上指定条件的控件，往指定方向滑动
void	<b>takeScreenshot</b> ( <i>String picName</i> ) 截图设备当前屏幕的图像并保存至 PC 上
void	<b>uninstall</b> ( <i>String packageName</i> ) 卸载设备上指定的应用程序
void	<b>wakeUp</b> () 唤醒设备

# Public Method

---

```
void assertActivity(String expected, Boolean result)
```

验证预期的 Activity 值是否与设备当前屏幕上的 Activity 相等

## Arguments

<i>expected</i>	预期设备屏幕当前的 Activity 值
<i>result</i>	True: 表示与预期值相等, False: 反之

```
void assertEquals(String actual, String expected, Boolean result)
```

验证实际值与预期的值是否相等

## Arguments

<i>actual</i>	实际文本值
<i>expected</i>	预期文本值
<i>result</i>	True: 表示与预期值相等, False: 反之

```
void assertExist(String type, string value, Int match)
```

验证设备屏幕上存在指定的控件

## Arguments

<i>type</i>	需要使用控件的哪种属性去查找控件, 具体值可参考 <a href="#">UIELEMENT</a>
<i>value</i>	具体的属性值, 详细参看 <code>uiautomatorviewer</code> 工具
<i>match</i>	整个页面相同条件的控件集合的下标值

```
void assertText(String expectText, String type, String value, Int match)
```

验证指定条件的控件 text 属性值与预期相等

## Arguments

<i>expectText</i>	预期指定控件的文本属性值
<i>type</i>	需要使用控件的哪种属性去查找控件, 具体值可参考 <a href="#">UIELEMENT</a>
<i>value</i>	具体的属性值, 详细参看 <code>uiautomatorviewer</code> 工具

*match*          整个页面相同条件的控件集合的下标值

```
void clearAppData(String packageName)
```

清除指定安装包的缓存数据

#### Arguments

*packageName*    应用程序的包名

```
void clearTextField(String type, String value, Int match)
```

清空指定条件的文本框内容

#### Arguments

*type*            需要使用控件的哪种属性去查找控件，具体值可参考 [UIELEMENT](#)

*value*           具体的属性值，详细参看 [uiautomatorviewer](#) 工具

*match*           整个页面相同条件的控件集合的下标值

```
void click( String type, String value, Int match)
```

通过指定条件在指定控件上发送一个点击事件

#### Arguments

*type*            需要使用控件的哪种属性去查找控件，具体值可参考 [UIELEMENT](#)

*value*           具体的属性值，详细参看 [uiautomatorviewer](#) 工具

*match*           整个页面相同条件的控件集合的下标值

```
void clickAndWaitForNewWindow( String type, String value, Int match, Int timeout)
```

在 *timeout* 时间内找到指定条件的控件，并在相应控件位置发送一个点击事件

#### Arguments

*type*            需要使用控件的哪种属性去查找控件，具体值可参考 [UIELEMENT](#)

*value*           具体的属性值，详细参看 [uiautomatorviewer](#) 工具

*match*           整个页面相同条件的控件集合的下标值

*timeout*        在 *timeout* 时间内找到控件，单位秒

```
void clickxy( String x, String y)
```

发送一个指定 x,y 坐标的点击事件到设备屏幕上

#### Arguments

<i>x</i>	想要点击屏幕某个位置的 X 坐标
<i>y</i>	想要点击屏幕某个位置的 Y 坐标

```
void closeApp(String packageName)
```

关闭设备上包名为 `packageName` 的应用程序

#### Arguments

`packageName` 应用程序的包名

```
void drag(String direction)
```

在设备屏幕上模拟一个滑动手势

#### Arguments

*direction* 指定一个滑动的方向，具体值可参考 [DIRECTION](#)

```
void dragxy(Int x, Int y, Int toX, Int toY)
```

通过指定起点坐标和终点坐标，在设备屏幕上模拟一个滑动手势

#### Arguments

<i>x</i>	滑动起点位置的 X 坐标
<i>y</i>	滑动起点位置的 Y 坐标
<i>toX</i>	滑动终点位置的 X 坐标
<i>toY</i>	滑动终点位置的 Y 坐标

```
void dragToEnd(String type, String value, Int match, Int steps)
```

向下滚动指定控件，直到控件无法向下滚动或者滚动次数达到 `steps` 值为止

#### Arguments

<i>type</i>	需要使用控件的哪种属性去查找控件，具体值可参考 <a href="#">UIELEMENT</a>
<i>value</i>	具体的属性值，详细参看 <code>uiautomatorviewer</code> 工具
<i>match</i>	整个页面相同条件的控件集合的下标值

*steps*            设置最大滚动的次数，如果滚动次数达到 *steps* 值，无论控件是否还可以继续滚动，操作都将停止

```
void dragToBegin(String type, String value, Int match, Int steps)
```

向上滚动指定控件，直到控件无法向上滚动或者滚动次数达到 *steps* 值为止

#### Arguments

*type*            需要使用控件的哪种属性去查找控件，具体值可参考 [UIELEMENT](#)

*value*           具体的属性值，详细参看 `uiautomatorviewer` 工具

*match*           整个页面相同条件的控件集合的下标值

*steps*           设置最大滚动的次数，如果滚动次数达到 *steps* 值，无论控件是否还可以继续滚动，操作都将停止

```
UiElement[] getAllElements()
```

获取设备当前屏幕上所有控件的集合

#### Return

[UiElement](#) 集合，一个 [UiElement](#) 对象包含了一个控件的所有属性

```
String getClassName (String type, String value, Int match)
```

通过指定条件的控件返回控件的 `class` 属性值

#### Arguments

*type*            需要使用控件的哪种属性去查找控件，具体值可参考 [UIELEMENT](#)

*value*           具体的属性值，详细参看 `uiautomatorviewer` 工具

*match*           整个页面相同条件的控件集合的下标值

#### Return

一个字符串类型的值，表示控件的 `class` 属性值

```
String getCurrentActivityName ()
```

获取设备当前屏幕的 `Activity` 值

#### Return

一个字符串类型的值，表示当前设备屏幕的 `Activity` 值

**UiElement** getElement (*String* type, *String* value, *Int* match)

返回指定条件的控件

#### Arguments

- type*            需要使用控件的哪种属性去查找控件，具体值可参考 [UIELEMENT](#)
- value*           具体的属性值，详细参看 `uiautomatorviewer` 工具
- match*           整个页面相同条件的控件集合的下标值

#### Return

一个 [UiElement](#) 的对象，包含着控件的所有属性值

**UiElement []** getElements (*String* type, *String* value, *Int* match)

返回设备当前屏幕上满足指定条件的所有控件集合

#### Arguments

- type*            需要使用控件的哪种属性去查找控件，具体值可参考 [UIELEMENT](#)
- value*           具体的属性值，详细参看 `uiautomatorviewer` 工具

#### Return

[UiElement](#) 集合，一个 [UiElement](#) 对象包含了一个控件的所有属性

**String** getSystemProp(*String* key)

获取设备的相关属性，等同于 `adb shell getprop <key>`

#### Arguments

- key*            设备属性参数，具体值可参考下面链接中的附录
- <http://developer.android.com/tools/help/MonkeyDevice.html>

#### Return

字符串类型值，具体数据值视 `key` 的值而定

**String** getText(*String* type, *String* value, *Int* match)

通过指定条件的控件返回控件的 `text` 属性值

#### Arguments

- type*            需要使用控件的哪种属性去查找控件，具体值可参考 [UIELEMENT](#)

<i>value</i>	具体的属性值，详细参看 <a href="#">uiautomatorviewer</a> 工具
<i>match</i>	整个页面相同条件的控件集合的下标值

### Return

一个字符串类型的值，表示控件的 **text** 属性值

```
(String,String) getXY (String type, String value, Int match)
```

通过指定条件的控件返回控件在设备屏幕上的坐标

### Arguments

<i>type</i>	需要使用控件的哪种属性去查找控件，具体值可参考 <a href="#">UIELEMENT</a>
<i>value</i>	具体的属性值，详细参看 <a href="#">uiautomatorviewer</a> 工具
<i>match</i>	整个页面相同条件的控件集合的下标值

### Return

返回两个值，第一个表示 X 坐标值，第二个表示 Y 坐标值

```
void input(String text, String type, String value, Int match)
```

在指定条件的控件上输入 **text** 值(注：如果 **type,value,match** 参数为空，则直接在被选中的文本框中输入 **text** 值)

### Arguments

<i>text</i>	想要输入的文本值
<i>type</i>	需要使用控件的哪种属性去查找控件，具体值可参考 <a href="#">UIELEMENT</a>
<i>value</i>	具体的属性值，详细参看 <a href="#">uiautomatorviewer</a> 工具
<i>match</i>	整个页面相同条件的控件集合的下标值

```
void install(String apkPath)
```

安装 **apkPath** 下的应用到设备中

### Arguments

<i>apkPath</i>	Apk 应用在 PC 上的位置
----------------	-----------------



```
void longclick(String type, String value, Int match)
```

在指定条件的控件上发送一个长按事件

#### Arguments

<i>type</i>	需要使用控件的哪种属性去查找控件，具体值可参考 <a href="#">UIELEMENT</a>
<i>value</i>	具体的属性值，详细参看 <a href="#">uiautomatorviewer</a> 工具
<i>match</i>	整个页面相同条件的控件集合的下标值

```
void press(String name)
```

发送一个 keycode 事件到设备中

#### Arguments

<i>name</i>	按键事件的名字，具体参数查看： <a href="http://developer.android.com/reference/android/view/KeyEvent.html">http://developer.android.com/reference/android/view/KeyEvent.html</a>
-------------	--

```
Popen shell(String cmd)
```

执行一条终端命令

#### Arguments

<i>cmd</i>	具体的执行命令
------------	---------

#### Return

一个 Popen 对象，包含命令执行结果信息

```
void sleep(Float seconds)
```

暂停执行脚本 seconds 时间

#### Arguments

<i>seconds</i>	脚本暂停执行的时间，单位为秒
----------------	----------------

```
void startActivity(String component)
```

启动设备上指定条件的应用程序

#### Arguments

<i>component</i>	参数格式为：应用程序包名/启动的 Activity 名
------------------	-----------------------------

```
void swipe(String direction,String type,String value,Int match,Int speed)
```

将设备屏幕上指定条件的控件，往指定方向滑动

#### Arguments

<i>direction</i>	滑动方向
<i>type</i>	需要使用控件的哪种属性去查找控件，具体值可参考 <a href="#">UIELEMENT</a>
<i>value</i>	具体的属性值，详细参看 <a href="#">uiautomatorviewer</a> 工具
<i>match</i>	整个页面相同条件的控件集合的下标值
<i>speed</i>	从滑动起点到滑动终点的时间差，数值越大滑动速度越慢

```
void takeScreenshot(String picName)
```

截图设备当前屏幕的图像并保存至 PC 上

#### Arguments

<i>picName</i>	图片保存的名字
----------------	---------

```
void uninstall(String packageName)
```

卸载设备上指定的应用程序

#### Arguments

<i>packageName</i>	卸载应用程序的包名
--------------------	-----------

```
void wakeUp()
```

唤醒设备

# UiElement

存储控件的所有属性信息

Methods	
String	getText() 控件 text 属性值
String	getClassName() 控件 class 属性值
String	getIndex() 控件 index 属性值
String	getPackage() 控件 package 属性值
String	getXY() 控件在设备屏幕上的坐标
Rect	getVisibleBounds() 控件的左上角坐标与右下角坐标
String	height() 控件在设备屏幕上的高度
String	width() 控件在设备屏幕上的宽度
Bool	isEnabled() 控件的 enabled 属性值
Bool	isScrollable() 控件的 scrollable 属性值
Bool	isChecked() 控件的 checked 属性值
Bool	isSelected() 控件的 selected 属性值
Bool	isFocused() 控件的 focused 属性值
String	sid() 控件的 resource-id 属性值（注：只能在 4.4 以上的系统使用）

# Rect

这个对象是用来存储成对出现的参数，比如，一个控件矩形框的左上角坐标、宽度和高度。

Methods	
String	<div>height() 控件 text 属性值</div>
String	<div>width() 控件 class 属性值</div>
Members	
String	<div>left 矩形框左边缘的 x 值</div>
String	<div>top 矩形框上边缘的 y 值</div>
String	<div>right 矩形框右边缘的 x 值</div>
String	<div>Bottom 矩形框下边缘的 y 值</div>

# 附录

## UIELEMENT

以控件的何种属性做为查找条件

Members	
TEXT	以控件的 <code>text</code> 属性做为查找条件
CLASSNAME	以控件的 <code>class</code> 属性做为查找条件
DESC	以控件的 <code>content-desc</code> 属性做为查找条件
SID	以控件的 <code>resource-id</code> 属性做为查找条件(注： 仅限 4.4 以上系统使用)

## DIRECTION

滑动方向

Members	
UP	向上滑动
DOWN	向下滑动
LEFT	向左滑动
RIGHT	向右滑动

## FLAG

关闭或者打开一些全局操作

NOTE: 脚本调试期间， 关闭一些全局操作可以提升脚本编写速度

Members	
SCREENSHOT	关闭脚本截图操作。 <b>False</b> : 关闭截图， <b>True</b> : 打开截图