

DGo Lab

# xDo 机器人使用手册

V1.51

Zeng Mary

2021-8-12

## 目录

APP 管理 .....	1
1. 安装 .....	1
2. 运行 .....	2
3. 停止 .....	2
屏幕管理 .....	2
1. 解锁 .....	2
2. 打开或关闭 .....	2
3. 获取并设置方向 .....	2
4. 截图 .....	2
控件操作 .....	3
1. 单击 .....	3
2. 双击 .....	3
3. 长按 .....	3
4. 拖拽 .....	3
5. 滑动 .....	3
6. 滚动 .....	4
7. 等待 .....	4
8. 剪贴板 .....	4
9. 多点手势 .....	4
10. 文本编辑 .....	5
11. 中心坐标 .....	5
元素查找、选择 .....	5
1. 常规 .....	5
2. 子孙 .....	5
3. 兄弟 .....	6
4. 方位 .....	6
5. 索引 .....	6
6. 状态 .....	6
7. 计数 .....	6
8. 属性 .....	6
9. XPath .....	6
10. 导出 (含属性、结构) .....	7
物理和虚拟按键操作 .....	7
输入法 .....	7

## APP 管理

### 1. 安装

`d.app_install('x.apk')` #仅限 URL 下载并安装

## 2. 运行

```
d.app_start("pkg_name")
d.app_start("pkg_name", " activity_name")
d.app_wait("pkg_name") #等待应用运行, 并返回进程 ID
d.app_wait("pkg_name", front=True) #等待应用前台运行
d.app_wait("pkg_name", timeout=20) #最长等待时间 20s (默认)
d.wait_activity("activity_name", timeout=10) #等待 Activity 出现时间 10s (默认)
```

## 3. 停止

```
d.app_stop("pkg_name")
d.app_clear("pkg_name")
```

## 屏幕管理

### 1. 解锁

```
d.unlock()
```

### 2. 打开或关闭

```
d.screen_on() #打开屏幕
d.screen_off() #关闭屏幕
d.info.get('screenOn') #获取屏幕当前状态
```

### 3. 获取并设置方向

```
d.orientation#获取设备当前方向
d.set_orientation('l') #设置设备方向 ("left")
d.set_orientation("r") #设置设备方向 ("right")
d.set_orientation("n") #设置设备方向 ("natural")
d.set_orientation("u") #设置设备方向 ("upside down")
d.freeze_rotation()# 禁止旋转
d.freeze_rotation(False)允许旋转
```

### 4. 截图

```
d.screenshot("xx.png") #截取设备当前图像
```

```
im = d(text="A").screenshot() #截取指定元素图像并保存  
im.save("xx.png")
```

## 控件操作

### 1. 单击

```
d.click(x, y)#单击  
d(text="A").click()  
d(text="A").click(timeout=10)  
d(text="A").click_exists(timeout=10) #等待元素出现，然后单击（限 10s）  
d(text="A").click_gone(maxretry=10, interval=1) #单击指定元素，直到消失为止，返回布尔值  
（期间最多尝试 10 次，间隔时长 1s）
```

### 2. 双击

```
d.double_click(x, y)#双击  
d.double_click(x, y, 0.1) #连续两次点击的间隔时间 0.1s（默认）
```

### 3. 长按

```
d(text="A").long_click()#长时间单击指定控件  
d.long_click(x, y)#长按  
d.long_click(x, y, 0.5) #点击时长持续 0.5s（默认）
```

### 4. 拖拽

```
d.drag(sx, sy, ex, ey) #拽  
d.drag(sx, sy, ex, ey, 0.5) #拽，耗时 0.5s（默认）  
d(text="A").drag_to(x, y, duration=0.5)#将 A 控件拖拽至 X,Y 坐标，耗时 0.5s  
d(text="A").drag_to(text="B", duration=0.25)#将控件 A 拖拽至 B 控件中心坐标，耗时 0.25s
```

### 5. 滑动

```
d.swipe(sx, sy, ex, ey)#滑动  
d.swipe(sx, sy, ex, ey, 0.5) #滑动持续时长 0.5s（默认）  
d.swipe_ext("right") #手指右滑，支持方位 "left", "right", "up", "down"  
d.swipe_ext("right", scale=0.9) #滑动幅度为屏幕宽度的 90%（默认）  
d.swipe_ext("right", box=(0, 0, 100, 100)) #在 (0,0) 至 (100, 100) 这个区域做滑动  
d.swipe_points([(x0, y0), (x1, y1), (x2, y2)], 0.2) #从 0 滑至 1，再到 2 坐标（支持比率），连续
```

两点操作耗时 0.2s (默认)

`d(text="A").swipe("right")` #滑动 A 控件至自身右边缘

`d(text="A").swipe("left", steps=10)` #同理, 1 步约为 5ms, 20 步约为 0.1s

`d(text="A").swipe("up", steps=20)`

`d(text="A").swipe("down", steps=20)`

## 6. 滚动

`d(scrollable=True).scroll(steps=20)` #垂直向前滚动 (默认), 等同于 `d(scrollable=True).scroll()`

`d(scrollable=True).scroll.backward()` #垂直向后滚动

`d(scrollable=True).scroll.toBeginning()` #垂直滚动至开始

`d(scrollable=True).scroll.toEnd()` #垂直滚动至末尾

`d(scrollable=True).scroll.to(text="A")` #垂直滚动至文本"A"

`d(scrollable=True).hscroll(steps=20)` #水平向前滚动 (默认)

`d(scrollable=True).hscroll.backward()`

`d(scrollable=True).hscroll.toBeginning()`

`d(scrollable=True).hscroll.toEnd()`

`d(scrollable=True).hscroll.to(text="A")`

附:

`d(scrollable=True).fling()` #垂直向前抛

`d(scrollable=True).fling.backward()` #垂直向后抛

`d(scrollable=True).fling.horiz.forward()` #水平向前抛

`d(scrollable=True).fling.horiz.toBeginning()` #水平抛至开始

`d(scrollable=True).fling.toEnd()` #垂直抛至结束

## 7. 等待

`d(text="A").wait(timeout=3.0)` #等待元素 A 出现, 限 3s 内返回布尔值

`d(text="A").wait_gone(timeout=1)` #等待元素消失, 限 1s 内

## 8. 剪贴板

`d.set_clipboard('text', 'label')` #设置剪贴板内容

`d.clipboard` #获取剪贴板内容

## 9. 多点手势

`d(text="A").gesture((sx1, sy1), (sx2, sy2), (ex1, ey1), (ex2, ey2))` #从一点到另一点的两点手势

`d(text="A").pinch_in(percent=100, steps=10)` #边缘到中心手势

`d(text="A").pinch_out()` #中心到边缘手势

## 10. 文本编辑

```
d(text="A").get_text() #获取文本
d(text="A").set_text("真是美好的一天") #设置文本
d(text="A").clear_text() #清除文本
```

## 11. 中心坐标

```
x, y = d(text="A").center() #返回元素"A"中心坐标
d(text="A").click(offset=(0.5, 0.5)) #点击元素"A"中心坐标, 等同于 d(text="A").click()
d(text="A").click(offset=(0, 0)) #单击元素"A"左上角坐标
d(text="A").click(offset=(1, 1)) #单击元素"A"右下角坐标
```

备注:

仿生模式下, 每次单击坐标将由系统随机设置, 而不是中心。

## 元素查找、选择

### 1. 常规

```
d(text='发送', className='android.widget.TextView') # 选择一个text为"发送", 且className为"android.widget.TextView"的控件
```

当前支持的类型如下

text	textContains	textMatches	textStartsWith
className	classNameMatches	description	descriptionContains
descriptionMatches	descriptionStartsWith	checkable	clickable
checked	longClickable	scrollable	enabled
focusable	focused	selected	packageName
packageNameMatches	resourceId	resourceIdMatches	index
instance			

### 2. 子孙

```
d(className="android.widget.ListView").child(text="查找") # 选择 className 为"android.widget.ListView"的一个子元素, 且子元素的 text 属性值为"查找"
```

### 3. 兄弟

`d(text="A").sibling(className="android.widget.ImageView")` #同上， 但为兄弟结点元素

### 4. 方位

`d(A).left(B)` #选择位于 A 左边的 B

`d(A).right(B)` #同理

`d(A).up(B)`

`d(A).down(B)`

### 5. 索引

`d(text="Add new", instance=0)` #当存在多个具有相同属性值的元素时， 选择第一个

`d(text="Add new")[0]` #当存在多个具有相同属性值的元素时 通过索引获取第一个

### 6. 状态

`d(text="A").exists` #检查 text 属性值为“A”的元素是否存在， 返回布尔值

`d(text="A").exists(timeout=3)` #同理， 但期限在 3 秒内

### 7. 计数

`d(text="Add new").count` #返回具有指定属性值元素的数量

### 8. 属性

`d(text="A").info` #查看指定元素的全部属性， 具体如下：

```
{ "index": "1", "text": " 文本 ", "resource-id": "pkg_name:id/qb", "package": "pkg_name", "content-desc": "", "checkable": "false", "checked": "false", "clickable": "false", "enabled": "true", "focusable": "false", "focused": "false", "scrollable": "false", "long-clickable": "false", "password": "false", "selected": "false", "visible-to-user": "true", "bounds": "[x1,y1][x2,y2]" }
```

### 9. XPath

常见用法

`xpath("//android.widget.TextView").wait(10)`

```
d.xpath("//*[@content-desc='文本']").click()
if d.xpath("//android.widget.TextView[contains(@text, '文本')]").exists():
    print("Yes")
for e in d.xpath("//android.widget.TextView").all():
    print(e.text)
```

备注：

更多用法，请参考元素“属性”部分

## 10. 导出（含属性、结构）

```
d.dump_hierarchy()
```

### 物理和虚拟按键操作

```
d.press("back") # 通过键名，点击返回键
d.press(0x07, 0x02) # 通过键码操作 0x07('O') with META ALT(0x02)
```

home	back	left	right
up	down	center	menu
search	enter	delete ( or del)	recent (recent apps)
volume_up	volume_down	volume_mute	camera
power			

当前所支持以上键名，如需使用键码，请查看安卓系统定义

### 输入法

针对未知控件的输入，使用方法如下：

```
d.set_fastinput_ime(True) #切换至 FastInputIME 输入法
d.send_keys("中文 ABC") #adb 广播输入“中文 ABC”
d.clear_text() # 清除输入框文本
d.set_fastinput_ime(False) #切换成正常的输入法
d.send_action("search") #输入法中的搜索，支持"search、end、ext、one、vious"
```

备注：

在 EditText 中输入内容后，若执行 press("enter")无反应，可使用 send\_action 函数解决