Appium\_Python\_Api文档

## 1.contexts

contexts(self):

Returns the contexts within the current session.

返回当前会话中的上下文，使用后可以识别H5页面的控件

:Usage:

driver.contexts

用法 driver.contexts

## 2. current\_context

current\_context(self):

Returns the current context of the current session.

返回当前会话的当前上下文

:Usage:

driver.current\_context

用法driver. current\_context

## 3. context

context(self):

Returns the current context of the current session.

返回当前会话的当前上下文。

:Usage:

driver.context

用法driver. Context

## 4. find\_element\_by\_ios\_uiautomation

find\_element\_by\_ios\_uiautomation(self, uia\_string):

Finds an element by uiautomation in iOS.

通过iOS uiautomation查找元素

:Args:

- uia\_string - The element name in the iOS UIAutomation library

:Usage:

driver.find\_element\_by\_ios\_uiautomation('.elements()[1].cells()[2]') 用法dr. find\_element\_by\_ios\_uiautomation(‘elements’)

## 5. find\_element\_by\_accessibility\_id

find\_element\_by\_accessibility\_id(self, id):

Finds an element by accessibility id.

通过accessibility id查找元素

:Args:

- id - a string corresponding to a recursive element search using the

Id/Name that the native Accessibility options utilize

:Usage:

driver.find\_element\_by\_accessibility\_id()

用法driver.find\_element\_by\_accessibility\_id(‘id’)

## 6.scroll

scroll(self, origin\_el, destination\_el):

Scrolls from one element to another

从元素origin\_el滚动至元素destination\_el

:Args:

- originalEl - the element from which to being scrolling - destinationEl - the element to scroll to

:Usage:

driver.scroll(el1, el2)

用法 driver.scroll(el1,el2)

## 7. drag\_and\_drop

drag\_and\_drop(self, origin\_el, destination\_el):

Drag the origin element to the destination element

将元素origin\_el拖到目标元素destination\_el

:Args:

- originEl - the element to drag

- destinationEl - the element to drag to

用法 driver.drag\_and\_drop(el1,el2)

## 8.tap

tap(self, positions, duration=None):

Taps on an particular place with up to five fingers, holding for a certain time

模拟手指点击（最多五个手指），可设置按住时间长度（毫秒）

:Args:

- positions - an array of tuples representing the x/y

coordinates of

the fingers to tap. Length can be up to five.

- duration - (optional) length of time to tap, in ms

:Usage:

driver.tap([(100, 20), (100, 60), (100, 100)], 500) 用法 driver.tap([(x,y),(x1,y1)],500)

## 9. swipe

swipe(self, start\_x, start\_y, end\_x, end\_y, duration=None):

Swipe from one point to another point, for an optional duration. 从A点滑动至B点，滑动时间为毫秒

:Args:

- start\_x - x-coordinate at which to start

- start\_y - y-coordinate at which to start

- end\_x - x-coordinate at which to stop

- end\_y - y-coordinate at which to stop

- duration - (optional) time to take the swipe, in ms.

:Usage:

driver.swipe(100, 100, 100, 400)

用法 driver.swipe(x1,y1,x2,y2,500)

## 10.flick

flick(self, start\_x, start\_y, end\_x, end\_y):

Flick from one point to another point.

按住A点后快速滑动至B点

:Args:

- start\_x - x-coordinate at which to start

- start\_y - y-coordinate at which to start

- end\_x - x-coordinate at which to stop

- end\_y - y-coordinate at which to stop

:Usage:

driver.flick(100, 100, 100, 400)

用法 driver.flick(x1,y1,x2,y2)

## 11.pinch

pinch(self, element=None, percent=200, steps=50):

Pinch on an element a certain amount

在元素上执行模拟双指捏（缩小操作）

:Args:

- element - the element to pinch

- percent - (optional) amount to pinch. Defaults to 200% - steps - (optional) number of steps in the pinch action

:Usage:

driver.pinch(element)

用法 driver.pinch(element)

## 12.zoom

zoom(self, element=None, percent=200, steps=50):

Zooms in on an element a certain amount

在元素上执行放大操作

:Args:

- element - the element to zoom

- percent - (optional) amount to zoom. Defaults to 200% - steps - (optional) number of steps in the zoom action

:Usage:

driver.zoom(element)

用法 driver.zoom(element)

## 13.reset

reset(self):

Resets the current application on the device.

重置应用(类似删除应用数据)

用法 driver.reset()

## 14. hide\_keyboard

hide\_keyboard(self, key\_name=None, key=None, strategy=None):

Hides the software keyboard on the device. In iOS, use `key\_name` to press a particular key, or `strategy`. In Android, no parameters are used.

隐藏键盘,iOS使用key\_name隐藏，安卓不使用参数

:Args:

- key\_name - key to press

- strategy - strategy for closing the keyboard (e.g.,

`tapOutside`)

driver.hide\_keyboard()

## 15. keyevent

keyevent(self, keycode, metastate=None):

Sends a keycode to the device. Android only. Possible keycodes can be found in

http://developer.android.com/reference/android/view/KeyEvent.html. 发送按键码（安卓仅有），按键码可以上网址中找到

:Args:

- keycode - the keycode to be sent to the device

- metastate - meta information about the keycode being sent 用法 dr.keyevent(‘4’)

## 16. press\_keycode

press\_keycode(self, keycode, metastate=None):

Sends a keycode to the device. Android only. Possible keycodes can be found in

http://developer.android.com/reference/android/view/KeyEvent.html. 发送按键码（安卓仅有），按键码可以上网址中找到

:Args:

- keycode - the keycode to be sent to the device

- metastate - meta information about the keycode being sent 用法 driver.press\_ keycode(‘4’)

dr.keyevent(‘4’)与driver.press\_ keycode(‘4’) 功能实现上一样的，都是按了返回键

## 17. long\_press\_keycode

long\_press\_keycode(self, keycode, metastate=None):

Sends a long press of keycode to the device. Android only. Possible keycodes can be

found in

http://developer.android.com/reference/android/view/KeyEvent.html. 发送一个长按的按键码（长按某键）

:Args:

- keycode - the keycode to be sent to the device

- metastate - meta information about the keycode being sent 用法 driver.long\_press\_keycode(‘4’)

## 18.current\_activity

current\_activity(self):

Retrieves the current activity on the device.

获取当前的activity

用法 print(driver.current\_activity())

## 19. wait\_activity

wait\_activity(self, activity, timeout, interval=1):

Wait for an activity: block until target activity presents or time out.

This is an Android-only method.

等待指定的activity出现直到超时，interval为扫描间隔1秒 即每隔几秒获取一次当前的activity

返回的True 或 False

:Agrs:

- activity - target activity

- timeout - max wait time, in seconds

- interval - sleep interval between retries, in seconds 用法driver.wait\_activity(‘.activity.xxx’,5,2)

## 20. background\_app

background\_app(self, seconds):

Puts the application in the background on the device for a certain duration.

后台运行app多少秒

:Args:

- seconds - the duration for the application to remain in the background

用法 driver.background\_app(5) 置后台5秒后再运行

## 21.is\_app\_installed

is\_app\_installed(self, bundle\_id):

Checks whether the application specified by `bundle\_id` is installed on the device.

检查app是否有安装

返回 True or False

:Args:

- bundle\_id - the id of the application to query

用法 driver.is\_app\_installed(“com.xxxx”)

## 22.install\_app

install\_app(self, app\_path):

Install the application found at `app\_path` on the device.

安装app,app\_path为安装包路径

:Args:

- app\_path - the local or remote path to the application to install

用法 driver.install\_app(app\_path)

## 23.remove\_app

remove\_app(self, app\_id):

Remove the specified application from the device.

删除app

:Args:

- app\_id - the application id to be removed

用法 driver.remove\_app(“com.xxx.”)

## 24.launch\_app

launch\_app(self):

Start on the device the application specified in the desired capabilities.

启动app

用法 driver.launch\_app()

## 25.close\_app

close\_app(self):

Stop the running application, specified in the desired capabilities, on the device.

关闭app

用法 driver.close\_app()

启动和关闭app运行好像会出错

## 26. start\_activity

start\_activity(self, app\_package, app\_activity, \*\*opts):

Opens an arbitrary activity during a test. If the activity belongs to another application, that application is started and the activity is opened.

This is an Android-only method.

在测试过程中打开任意活动。如果活动属于另一个应用程序，该应用程序的启动和活动被打开。

这是一个安卓的方法

:Args:

- app\_package - The package containing the activity to start. - app\_activity - The activity to start.

- app\_wait\_package - Begin automation after this package starts (optional).

- app\_wait\_activity - Begin automation after this activity starts (optional).

- intent\_action - Intent to start (optional).

- intent\_category - Intent category to start (optional). - intent\_flags - Flags to send to the intent (optional).

- optional\_intent\_arguments - Optional arguments to the intent (optional).

- stop\_app\_on\_reset - Should the app be stopped on reset (optional)?

用法 driver.start\_activity(app\_package, app\_activity)

## 27.lock

lock(self, seconds):

Lock the device for a certain period of time. iOS only.

锁屏一段时间 iOS专有

:Args:

- the duration to lock the device, in seconds

用法 driver.lock()

## 28.shake

shake(self):

Shake the device.

摇一摇手机

用法 driver.shake()

29.open\_notifications

open\_notifications(self):

Open notification shade in Android (API Level 18 and above) 打系统通知栏（仅支持API 18 以上的安卓系统）

用法 driver.open\_notifications()

## 30.network\_connection

network\_connection(self):

Returns an integer bitmask specifying the network connection type. Android only.

返回网络类型 数值

Possible values are available through the enumeration

`appium.webdriver.ConnectionType`

用法 driver.network\_connection

## 31. set\_network\_connection

set\_network\_connection(self, connectionType):

Sets the network connection type. Android only.

Possible values:

Value (Alias) | Data | Wifi | Airplane Mode

-------------------------------------------------

0 (None) | 0 | 0 | 0

1 (Airplane Mode) | 0 | 0 | 1

2 (Wifi only) | 0 | 1 | 0

4 (Data only) | 1 | 0 | 0

6 (All network on) | 1 | 1 | 0

These are available through the enumeration

`appium.webdriver.ConnectionType`

设置网络类型

:Args:

- connectionType - a member of the enum

appium.webdriver.ConnectionType

用法 先加载from appium.webdriver.connectiontype import

ConnectionType

dr.set\_network\_connection(ConnectionType.WIFI\_ONLY)

ConnectionType的类型有

NO\_CONNECTION = 0

AIRPLANE\_MODE = 1

WIFI\_ONLY = 2

DATA\_ONLY = 4

ALL\_NETWORK\_ON = 6

## 32. available\_ime\_engines

available\_ime\_engines(self):

Get the available input methods for an Android device. Package and activity are returned (e.g.,

['com.android.inputmethod.latin/.LatinIME'])

Android only.

返回安卓设备可用的输入法

用法print(driver.available\_ime\_engines)

## 33.is\_ime\_active

is\_ime\_active(self):

Checks whether the device has IME service active. Returns True/False.

Android only.

检查设备是否有输入法服务活动。返回真/假。

安卓

用法 print(driver.is\_ime\_active())

## 34.activate\_ime\_engine

activate\_ime\_engine(self, engine):

Activates the given IME engine on the device.

Android only.

激活安卓设备中的指定输入法，设备可用输入法可以从

“available\_ime\_engines”获取

:Args:

- engine - the package and activity of the IME engine to activate (e.g.,

'com.android.inputmethod.latin/.LatinIME')

用法

driver.activate\_ime\_engine(“com.android.inputmethod.latin/.LatinIME”)

## 35.deactivate\_ime\_engine

deactivate\_ime\_engine(self):

Deactivates the currently active IME engine on the device.

Android only.

关闭安卓设备当前的输入法

用法 driver.deactivate\_ime\_engine()

## 36.active\_ime\_engine

active\_ime\_engine(self):

Returns the activity and package of the currently active IME engine (e.g.,

'com.android.inputmethod.latin/.LatinIME').

Android only.

返回当前输入法的包名

用法 driver.active\_ime\_engine

## 37. toggle\_location\_services

toggle\_location\_services(self):

Toggle the location services on the device. Android only.

打开安卓设备上的位置定位设置

用法 driver.toggle\_location\_services()

## 38.set\_location

set\_location(self, latitude, longitude, altitude):

Set the location of the device

设置设备的经纬度

:Args:

- latitude纬度 - String or numeric value between -90.0 and 90.00

- longitude经度 - String or numeric value between -180.0 and 180.0

- altitude海拔高度- String or numeric value

用法 driver.set\_location(纬度，经度，高度)

## 39.tag\_name

tag\_name(self):

This element's ``tagName`` property.

返回元素的tagName属性

经实践返回的是class name

用法 element.tag\_name()

## 40.text

text(self):

The text of the element.

返回元素的文本值

用法 element.text()

## 41.click

click(self):

Clicks the element.

点击元素

用法 element.click()

## 42.submit

submit(self):

Submits a form.

提交表单

用法 暂无

## 43.clear

clear(self):

Clears the text if it's a text entry element.

清除输入的内容

用法 element.clear()

## 44.get\_attribute

get\_attribute(self, name):

详见@chenhengjie123 的超级链接

Gets the given attribute or property of the element.

1、获取 content-desc 的方法为 get\_attribute("name") ，而且还不能保证返回的一定是 content-desc （content-desc 为空时会返回 text 属性值）

2、get\_attribute 方法不是我们在 uiautomatorviewer 看到的所有属性都能获取的（此处的名称均为使用 get\_attribute 时使用的属性名称）： 可获取的：

字符串类型：

name(返回 content-desc 或 text)

text(返回 text)

className(返回 class，只有 API=>18 才能支持)

resourceId(返回 resource-id，只有 API=>18 才能支持)

This method will first try to return the value of a property with the

given name. If a property with that name doesn't exist, it returns the

value of the attribute with the same name. If there's no attribute with

that name, ``None`` is returned.

Values which are considered truthy, that is equals "true" or "false",

are returned as booleans. All other non-``None`` values are returned

as strings. For attributes or properties which do not exist, ``None``

is returned.

:Args:

- name - Name of the attribute/property to retrieve.

Example::

# Check if the "active" CSS class is applied to an element. is\_active = "active" in target\_element.get\_attribute("class") 用法 暂无

## 45.is\_selected

is\_selected(self):

Returns whether the element is selected.

Can be used to check if a checkbox or radio button is selected. 返回元素是否选择。

可以用来检查一个复选框或单选按钮被选中。

用法 element.is\_slected()

## 46.is\_enabled

is\_enabled(self):

Returns whether the element is enabled.

返回元素是否可用True of False

用法 element.is\_enabled()

## 47.find\_element\_by\_id

find\_element\_by\_id(self, id\_):

Finds element within this element's children by ID.

通过元素的ID定位元素

:Args:

- id\_ - ID of child element to locate.

用法 driver. find\_element\_by\_id(“id”)

## 48. find\_elements\_by\_id

find\_elements\_by\_id(self, id\_):

Finds a list of elements within this element's children by ID. 通过元素ID定位,含有该属性的所有元素

:Args:

- id\_ - Id of child element to find.

用法 driver. find\_elements\_by\_id(“id”)

## 49. find\_element\_by\_name

find\_element\_by\_name(self, name):

Finds element within this element's children by name.

通过元素Name定位（元素的名称属性text）

:Args:

- name - name property of the element to find.

用法 driver.find\_element\_by\_name(“name”)

## 50. find\_elements\_by\_name

find\_elements\_by\_name(self, name):

Finds a list of elements within this element's children by name. 通过元素Name定位（元素的名称属性text），含有该属性的所有元素 :Args:

- name - name property to search for.

用法 driver.find\_element\_by\_name(“name”)

## 51. find\_element\_by\_link\_text

find\_element\_by\_link\_text(self, link\_text):

Finds element within this element's children by visible link text. 通过元素可见链接文本定位

:Args:

- link\_text - Link text string to search for.

用法 driver.find\_element\_by\_link\_text(“text”)

## 52. find\_elements\_by\_link\_text

find\_element\_by\_link\_text(self, link\_text):

Finds a list of elements within this element's children by visible link text

通过元素可见链接文本定位,含有该属性的所有元素

:Args:

- link\_text - Link text string to search for.

用法 driver.find\_elements\_by\_link\_text(“text”)

## 53. find\_element\_by\_partial\_link\_text

find\_element\_by\_partial\_link\_text(self, link\_text):

Finds element within this element's children by partially visible link text.

通过元素部分可见链接文本定位

:Args:

- link\_text - Link text string to search for.

driver. find\_element\_by\_partial\_link\_text(“text”)

## 54. find\_elements\_by\_partial\_link\_text

find\_elements\_by\_partial\_link\_text(self, link\_text):

Finds a list of elements within this element's children by link text. 通过元素部分可见链接文本定位,含有该属性的所有元素

:Args:

- link\_text - Link text string to search for.

driver. find\_elements\_by\_partial\_link\_text(“text”)

## 55. find\_element\_by\_tag\_name

find\_element\_by\_tag\_name(self, name):

Finds element within this element's children by tag name.

通过查找html的标签名称定位元素

:Args:

- name - name of html tag (eg: h1, a, span)

用法 driver.find\_element\_by\_tag\_name(“name”)

## 56. find\_elements\_by\_tag\_name

find\_elements\_by\_tag\_name(self, name):

Finds a list of elements within this element's children by tag name. 通过查找html的标签名称定位所有元素

:Args:

- name - name of html tag (eg: h1, a, span)

用法driver.find\_elements\_by\_tag\_name(“name”)

## 57. find\_element\_by\_xpath

find\_element\_by\_xpath(self, xpath):

Finds element by xpath.

通过Xpath定位元素，详细方法可参阅

http://www.w3school.com.cn/xpath/

:Args:

xpath - xpath of element to locate.

"//input[@class='myelement']"

Note: The base path will be relative to this element's location.

This will select the first link under this element.

::

myelement.find\_elements\_by\_xpath(".//a")

However, this will select the first link on the page.

::

myelement.find\_elements\_by\_xpath("//a")

用法 find\_element\_by\_xpath(“//\*”)

## 58. find\_elements\_by\_xpath

find\_elements\_by\_xpath(self, xpath):

Finds elements within the element by xpath.

:Args:

- xpath - xpath locator string.

Note: The base path will be relative to this element's location.

This will select all links under this element.

::

myelement.find\_elements\_by\_xpath(".//a")

However, this will select all links in the page itself.

::

myelement.find\_elements\_by\_xpath("//a")

用法find\_elements\_by\_xpath(“//\*”)

## 59. find\_element\_by\_class\_name

find\_element\_by\_class\_name(self, name):

Finds element within this element's children by class name. 通过元素class name属性定位元素

:Args:

- name - class name to search for.

用法 driver.

find\_element\_by\_class\_name(“android.widget.LinearLayout”)

## 60. find\_elements\_by\_class\_name

find\_elements\_by\_class\_name(self, name):

Finds a list of elements within this element's children by class name.

通过元素class name属性定位所有含有该属性的元素

:Args:

- name - class name to search for.

用法 driver.

find\_elements\_by\_class\_name(“android.widget.LinearLayout”)

## 61. find\_element\_by\_css\_selector

find\_element\_by\_css\_selector(self, css\_selector):

Finds element within this element's children by CSS selector. 通过CSS选择器定位元素

:Args:

- css\_selector - CSS selctor string, ex: 'a.nav#home'

## 62.send\_keys

send\_keys(self, \*value):

Simulates typing into the element.

在元素中模拟输入（开启appium自带的输入法并配置了appium输入法后，可以输入中英文）

:Args:

- value - A string for typing, or setting form fields. For setting

file inputs, this could be a local file path.

Use this to send simple key events or to fill out form fields::

form\_textfield = driver.find\_element\_by\_name('username') form\_textfield.send\_keys("admin")

This can also be used to set file inputs.

::

file\_input = driver.find\_element\_by\_name('profilePic')

file\_input.send\_keys("path/to/profilepic.gif")

# Generally it's better to wrap the file path in one of the methods

# in os.path to return the actual path to support cross OS testing.

#

file\_input.send\_keys(os.path.abspath("path/to/profilepic.gif")) driver.element.send\_keys(“中英”)

## 63. is\_displayed

is\_displayed(self):

Whether the element is visible to a user.

此元素用户是否可见。简单地说就是隐藏元素和被控件挡住无法操作的元素（仅限 Selenium，appium是否实现了类似功能不是太确定）这一项都会返回 False

用法 driver.element.is\_displayed()

## 64. location\_once\_scrolled\_into\_view

location\_once\_scrolled\_into\_view(self):

"""THIS PROPERTY MAY CHANGE WITHOUT WARNING. Use this to discover where on the screen an element is so that we can click it. This method

should cause the element to be scrolled into view.

Returns the top lefthand corner location on the screen, or ``None`` if

the element is not visible.

暂不知道用法

"""

## 65.size

size(self):

The size of the element.

获取元素的大小（高和宽）

new\_size["height"] = size["height"]

new\_size["width"] = size["width"]

用法 driver.element.size

## 66. value\_of\_css\_property

value\_of\_css\_property(self, property\_name):

The value of a CSS property.

CSS属性

用法 暂不知

## 67.location

location(self):

The location of the element in the renderable canvas.

获取元素左上角的坐标

用法 driver.element.location

'''返回element的x坐标, int类型'''

driver.element.location.get('x')

'''返回element的y坐标, int类型'''

driver.element.location.get('y')

## 68.rect

rect(self):

A dictionary with the size and location of the element.

元素的大小和位置的字典

## 69. screenshot\_as\_base64

screenshot\_as\_base64(self):

Gets the screenshot of the current element as a base64 encoded string.

获取当前元素的截图为Base64编码的字符串

:Usage:

img\_b64 = element.screenshot\_as\_base64

## 70.execute\_script

execute\_script(self, script, \*args):

Synchronously Executes JavaScript in the current window/frame. 在当前窗口/框架（特指 Html 的 iframe ）同步执行 javascript 代码。你可以理解为如果这段代码是睡眠5秒，这五秒内主线程的 javascript 不会执行 :Args:

- script: The JavaScript to execute.

- \\*args: Any applicable arguments for your JavaScript.

:Usage:

driver.execute\_script('document.title')

## 71.execute\_async\_script

execute\_async\_script(self, script, \*args):

Asynchronously Executes JavaScript in the current window/frame. 插入 javascript 代码，只是这个是异步的，也就是如果你的代码是睡眠5秒，那么你只是自己在睡，页面的其他 javascript 代码还是照常执行 :Args:

- script: The JavaScript to execute.

- \\*args: Any applicable arguments for your JavaScript.

:Usage:

driver.execute\_async\_script('document.title')

## 72.current\_url

current\_url(self):

Gets the URL of the current page.

获取当前页面的网址。

:Usage:

driver.current\_url

用法 driver.current\_url

## 73. page\_source

page\_source(self):

Gets the source of the current page.

获取当前页面的源。

:Usage:

driver.page\_source

## 74.close

close(self):

Closes the current window.

关闭当前窗口

:Usage:

driver.close()

## 75.quit

quit(self):

Quits the driver and closes every associated window.

退出脚本运行并关闭每个相关的窗口连接 :Usage:

driver.quit()