

---

# uiautomator2

Feb 18, 2020



---

## Contents:

---

<b>1</b>	<b>API</b>	<b>1</b>
<b>2</b>	<b>Indices and tables</b>	<b>9</b>
	<b>Index</b>	<b>11</b>



```
class uiautomator2.Device (host='127.0.0.1', port=7912)
```

```
    adb_shell (*args)
```

**Example:** `adb_shell('pwd')` `adb_shell('ls', '-l')` `adb_shell('ls -l')`

**Returns:** string for stdout merged with stderr, after the entire shell command is completed.

```
    app_clear (pkg_name)
```

Stop and clear app data: `pm clear`

```
    app_current ()
```

**Returns:** dict(package, activity, pid?)

**Raises:** `EnvironmentError`

**For developer:** Function `reset_uiautomator` need this function, so can't use jsonrpc here.

```
    app_icon (package_name: str)
```

**Returns:** `PIL.Image`

**Raises:** `UiaError`

```
    app_info (pkg_name)
```

Get app info

**Args:** `pkg_name (str)`: package name

**Return example:**

```
    { "mainActivity": "com.github.uiautomator.MainActivity", "label": "ATX", "versionName":  
      "1.1.7", "versionCode": 1001007, "size": 1760809  
    }
```

**Raises:** `UiaError`

**app\_install** (*url*, *installing\_callback=None*, *server=None*)  
{u'message': u'downloading', "progress": {u'totalSize': 407992690, u'copiedSize': 49152}}

**Returns:** packageName

**Raises:** RuntimeError

**app\_list** (*filter: str = None*) → list

**Args:** filter: [-f] [-d] [-e] [-s] [-3] [-i] [-u] [--user USER\_ID] [FILTER]

**Returns:** list of apps by filter

**app\_list\_running** () → list

**Returns:** list of running apps

**app\_start** (*package\_name*, *activity=None*, *extras={}*, *wait=False*, *stop=False*, *unlock=False*, *launch\_timeout=None*, *use\_monkey=False*)  
Launch application Args:

package\_name (str): package name activity (str): app activity stop (bool): Stop app before starting the activity. (require activity) use\_monkey (bool): use monkey command to start app when activity is not given wait (bool): wait until app started. default False

**Raises:** SessionBrokenError

**app\_stop** (*pkg\_name*)  
Stop one application: am force-stop

**app\_stop\_all** (*excludes=[]*)  
Stop all third party applications Args:

excludes (list): apps that do now want to kill

**Returns:** a list of killed apps

**app\_uninstall** (*pkg\_name*) → bool  
Uninstall an app

**Returns:** bool: success

**app\_uninstall\_all** (*excludes=[]*, *verbose=False*)  
Uninstall all apps

**app\_wait** (*package\_name: str*, *timeout: float = 20.0*, *front=False*) → int  
Wait until app launched Args:

package\_name (str): package name timeout (float): maxium wait time front (bool): wait until app is current app

**Returns:** pid (int) 0 if launch failed

**disable\_popups** (*enable=True*)  
Automatic click all popups TODO: need fix

**healthcheck** ()  
Reset device into health state

**Raises:** RuntimeError

**hooks\_apply** (*stage*, *func\_name*, *args=()*, *kwargs={}*, *ret=None*)

**Args:** stage(str): one of "before" or "after"

**hooks\_register** (*func*)

**Args:** func: should accept 3 args. func\_name:string, args:tuple, kwargs:dict

**jsonrpc**

Make jsonrpc call easier For example:

```
self.jsonrpc.pressKey("home")
```

**jsonrpc\_call** (*method, params=[], http\_timeout=60*)

jsonrpc2 call Refs:

- <http://www.jsonrpc.org/specification>

**open\_identify** (*theme='black'*)

**Args:** theme (str): black or red

**pull** (*src: str, dst: str*)

Pull file from device to local

**Raises:** FileNotFoundError(py3) OSError(py2)

Require atx-agent >= 0.0.9

**pull\_content** (*src: str*) → bytes

Read remote file content

**Raises:** FileNotFoundError

**push** (*src, dst, mode=420*)

**Args:** src (path or fileobj): source file dst (str): destination can be folder or file path

**Returns:** dict object, for example:

```
{"mode": "0660", "size": 63, "target": "/sdcard/ABOUT.rst"}
```

Since chmod may fail in android, the result "mode" may not same with input args(mode)

**Raises:** IOError(if push got something wrong)

**push\_url** (*url, dst, mode=420*)

**Args:** url (str): http url address dst (str): destination mode (str): file mode

**Raises:** FileNotFoundError(py3) OSError(py2)

**request\_agent** (*relative\_url: str, method='get', timeout=60.0*)

send http-request to atx-agent

**reset\_uiautomator** (*reason='unknown'*)

Reset uiautomator

**Raises:** RuntimeError

**Orders:**

- stop uiautomator keeper
- am force-stop com.github.uiautomator
- start uiautomator keeper(am instrument -w ...)
- wait until uiautomator service is ready

**screenshot** (*\*args, \*\*kwargs*)

Take screenshot of device

**Returns:** PIL.Image

**service** (*name*)

Manage service start or stop

**Example:** d.service("uiautomator").start() d.service("uiautomator").stop()

**session** (*pkg\_name=None, attach=False, launch\_timeout=None, strict=False*)

Create a new session

**Args:** pkg\_name (str): android package name attach (bool): attach to already running app launch\_timeout (int): launch timeout strict (bool): used along with attach,

when attach and strict both true, SessionBrokenError will raise if app not running

**Raises:** requests.HTTPError, SessionBrokenError

**set\_new\_command\_timeout** (*timeout: int*)

default 3 minutes Args:

timeout (int): seconds

**setup\_jsonrpc** (*jsonrpc\_url=None*)

Wrap jsonrpc call into object Usage example:

self.setup\_jsonrpc().pressKey("home")

**shell** (*cmdargs, stream=False, timeout=60*)

Run adb shell command with arguments and return its output. Require atx-agent >=0.3.3

**Args:** cmdargs: str or list, example: "ls -l" or ["ls", "-l"] timeout: seconds of command run, works on when stream is False stream: bool used for long running process.

**Returns:** (output, exit\_code) when stream is False requests.Response when stream is True, you have to close it after using

**Raises:** RuntimeError

For atx-agent is not support return exit code now. When command got something wrong, exit\_code is always 1, otherwise exit\_code is always 0

**unlock** ()

unlock screen

**wait\_activity** (*activity, timeout=10*)

wait activity Args:

activity (str): name of activity timeout (float): max wait time

**Returns:** bool of activity

**window\_size** ()

return (width, height)

**class** uiautomator2.Session (*server, pkg\_name=None, pid=None*)

**clear\_text** ()

clear text Raises:

EnvironmentError

**clear\_traversed\_text** ()

clear the last traversed text.



**click** (*x, y*)  
click position

**close** ()  
close app

**current\_ime** ()  
Current input method Returns:  
(method\_id(str), shown(bool))

**Example output:** ("com.github.uiautomator/.FastInputIME", True)

**double\_click** (*x, y, duration=0.1*)  
double click position

**drag** (*sx, sy, ex, ey, duration=0.5*)  
Swipe from one point to another point.

**dump\_hierarchy** (*compressed=False, pretty=False*) → str

**Args:** shell (bool): use "adb shell uiautomator dump" to get hierarchy pretty (bool): format xml

**Same as** content = self.jsonrpc.dumpWindowHierarchy(compressed, None)

But through GET /dump/hierarchy will be more robust when dumpHierarchy fails, the atx-agent will restart uiautomator again, then retry

v-1.3.4 change back to jsonrpc.dumpWindowHierarchy

**freeze\_rotation** (*freeze=True*)  
freeze or unfreeze the device rotation in current status.

**implicitly\_wait** (*seconds=None*)  
set default wait timeout Args:  
seconds(float): to wait element show up

**Deprecated:** recommend use: d.settings['wait\_timeout'] = 10

**last\_traversed\_text**  
get last traversed text. used in webview for highlighted text.

**long\_click** (*x, y, duration=None*)  
long click at arbitrary coordinates. Args:  
duration (float): seconds of pressed

**make\_toast** (*text, duration=1.0*)  
Show toast Args:  
text (str): text to show duration (float): seconds of display

**orientation**  
orienting the devie to left/right or natural. left/l: rotation=90 , displayRotation=1 right/r: rotation=270, displayRotation=3 natural/n: rotation=0 , displayRotation=0 upsidedown/u: rotation=180, displayRotation=2

**press** (*key, meta=None*)

**press key via name or key code. Supported key name includes:** home, back, left, right, up, down, center, menu, search, enter, delete(or del), recent(recent apps), volume\_up, volume\_down, volume\_mute, camera, power.

**restart** (*use\_monkey=False*)

Stop app and start

**Raises:** RuntimeError

**running** ()

Check is session is running. return bool

**screenshot** (*filename=None, format='pillow'*)

Image format is JPEG

**Args:** filename (str): saved filename format (string): used when filename is empty. one of “pillow” or “opencv”

**Raises:** IOError, SyntaxError

**Examples:** screenshot(“saved.jpg”) screenshot().save(“saved.png”) cv2.imwrite(‘saved.jpg’, screenshot(format=‘opencv’))

**send\_action** (*code*)

Simulate input method edito code

**Args:** code (str or int): input method editor code

**Examples:** send\_action(“search”), send\_action(3)

**Refs:** <https://developer.android.com/reference/android/view/inputmethod/EditorInfo>

**send\_keys** (*text: str, clear: bool = False*)

**Args:** text (str): text to set clear (bool): clear before set text

**Raises:** EnvironmentError

**set\_clipboard** (*text, label=None*)

**Args:** text: The actual text in the clip. label: User-visible label for the clip data.

**set\_fastinput\_ime** (*enable=True*)

Enable of Disable FastInputIME

**set\_orientation** (*value*)

setter of orientation property.

**swipe** (*fx, fy, tx, ty, duration=0.1, steps=None*)

**Args:** fx, fy: from position tx, ty: to position duration (float): duration steps: 1 steps is about 5ms, if set, duration will be ignore

**Documents:** uiautomator use steps instead of duration As the document say: Each step execution is throttled to 5ms per step.

**Links:** <https://developer.android.com/reference/android/support/test/uiautomator/UiDevice.html#swipe%28int,%20int,%20int,%20int,%20int%29>

**swipe\_points** (*points, duration=0.5*)

**Args:** points: is point array containg at least one point object. eg [[200, 300], [210, 320]] duration: duration to inject between two points

**Links:** [https://developer.android.com/reference/android/support/test/uiautomator/UiDevice.html#swipe\(android.graphics.Point\[\],int\)](https://developer.android.com/reference/android/support/test/uiautomator/UiDevice.html#swipe(android.graphics.Point[],int))

**tap** (*x, y*)

alias of click

**touch**

ACTION\_DOWN: 0 ACTION\_MOVE: 2 touch.down(x, y) touch.move(x, y) touch.up()

**wait\_fastinput\_ime** (*timeout=5.0*)

wait FastInputIME is ready Args:

timeout(float): maxium wait time

**Raises:** EnvironmentError



## CHAPTER 2

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



## A

adb\_shell() (*uiautomator2.Device method*), 1  
 app\_clear() (*uiautomator2.Device method*), 1  
 app\_current() (*uiautomator2.Device method*), 1  
 app\_icon() (*uiautomator2.Device method*), 1  
 app\_info() (*uiautomator2.Device method*), 1  
 app\_install() (*uiautomator2.Device method*), 1  
 app\_list() (*uiautomator2.Device method*), 2  
 app\_list\_running() (*uiautomator2.Device method*), 2  
 app\_start() (*uiautomator2.Device method*), 2  
 app\_stop() (*uiautomator2.Device method*), 2  
 app\_stop\_all() (*uiautomator2.Device method*), 2  
 app\_uninstall() (*uiautomator2.Device method*), 2  
 app\_uninstall\_all() (*uiautomator2.Device method*), 2  
 app\_wait() (*uiautomator2.Device method*), 2

## C

clear\_text() (*uiautomator2.Session method*), 4  
 clear\_traversed\_text() (*uiautomator2.Session method*), 4  
 click() (*uiautomator2.Session method*), 4  
 close() (*uiautomator2.Session method*), 5  
 current\_time() (*uiautomator2.Session method*), 5

## D

Device (*class in uiautomator2*), 1  
 disable\_popups() (*uiautomator2.Device method*), 2  
 double\_click() (*uiautomator2.Session method*), 5  
 drag() (*uiautomator2.Session method*), 5  
 dump\_hierarchy() (*uiautomator2.Session method*), 5

## F

freeze\_rotation() (*uiautomator2.Session method*), 5

## H

healthcheck() (*uiautomator2.Device method*), 2  
 hooks\_apply() (*uiautomator2.Device method*), 2  
 hooks\_register() (*uiautomator2.Device method*), 3

## I

implicitly\_wait() (*uiautomator2.Session method*), 5

## J

jsonrpc (*uiautomator2.Device attribute*), 3  
 jsonrpc\_call() (*uiautomator2.Device method*), 3

## L

last\_traversed\_text (*uiautomator2.Session attribute*), 5  
 long\_click() (*uiautomator2.Session method*), 5

## M

make\_toast() (*uiautomator2.Session method*), 5

## O

open\_identify() (*uiautomator2.Device method*), 3  
 orientation (*uiautomator2.Session attribute*), 5

## P

press() (*uiautomator2.Session method*), 5  
 pull() (*uiautomator2.Device method*), 3  
 pull\_content() (*uiautomator2.Device method*), 3  
 push() (*uiautomator2.Device method*), 3  
 push\_url() (*uiautomator2.Device method*), 3

## R

request\_agent() (*uiautomator2.Device method*), 3  
 reset\_uiautomator() (*uiautomator2.Device method*), 3  
 restart() (*uiautomator2.Session method*), 5  
 running() (*uiautomator2.Session method*), 6

## S

`screenshot()` (*uiautomator2.Device method*), 3  
`screenshot()` (*uiautomator2.Session method*), 6  
`send_action()` (*uiautomator2.Session method*), 6  
`send_keys()` (*uiautomator2.Session method*), 6  
`service()` (*uiautomator2.Device method*), 4  
`Session` (class in *uiautomator2*), 4  
`session()` (*uiautomator2.Device method*), 4  
`set_clipboard()` (*uiautomator2.Session method*), 6  
`set_fastinput_ime()` (*uiautomator2.Session method*), 6  
`set_new_command_timeout()` (*uiautomator2.Device method*), 4  
`set_orientation()` (*uiautomator2.Session method*), 6  
`setup_jsonrpc()` (*uiautomator2.Device method*), 4  
`shell()` (*uiautomator2.Device method*), 4  
`swipe()` (*uiautomator2.Session method*), 6  
`swipe_points()` (*uiautomator2.Session method*), 6

## T

`tap()` (*uiautomator2.Session method*), 6  
`touch` (*uiautomator2.Session attribute*), 6

## U

`unlock()` (*uiautomator2.Device method*), 4

## W

`wait_activity()` (*uiautomator2.Device method*), 4  
`wait_fastinput_ime()` (*uiautomator2.Session method*), 7  
`window_size()` (*uiautomator2.Device method*), 4