
Mykonos

Release 1.4.7

Sep 27, 2019

1	Introduction	1
2	Installation	3
3	Usage	5
4	Selector Support:	7
4.1	Mykonos Keywords	7
5	Indices and tables	19
	Python Module Index	21
	Index	23

CHAPTER 1

Introduction

Mykonos is a complete test automation tools for Android Device using Robot Framework and UI Automator (Python), it easy to learn because Mykonos use BDD syntax to write the test cases.

CHAPTER 2

Installation

`pip install mykonos`

CHAPTER 3

Usage

- Download and Install android emulator, base on the guidance [Genymotion](#) or [Android Emulator](#).
- Make sure emulator is available by checking with *adb devices*, for more detail info please check the adb command on [ADB Shell](#) guidance.
- Make sure Robot Framework is able to run by execute *robot -version* and it will get Robot Framework version as a result.
- Create a file (**sample.robot**).
- Import **__Mykonos__** Library on the Robot Framework Test Suite.
- Write test case base on [Robot Framework](#) guidance.

Selector Support:

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

4.1 Mykonos Keywords

class mykonos.keywords.element.Click

click_a_point (*device=None, *argument, **settings*)

Click into pointer target location.

This keyword is used to click location based on pointer X and Y.

Example:

|| Click A Point |x=10 |y=20

With Device/ Pararel : || @{emulator} = | 192.168.1.1 | 192.168.1.2 || Click A Point |x=10 |y=20 | device_parallel=@{emulator}

click_element (*device=None, *argument, **settings*)

Click on UI base on locator.

This keyword is used to click button or element of device.

Example:

|| Click Element | className=sample class

With Device/ Pararel : || @ {emulator} = | 192.168.1.1 | 192.168.1.2 || Click Element | className=sample class | `device_parallel=@ {emulator}`

long_click_element (*device=None, *argument, **settings*)

Long click on UI base on locator.

This keyword is used to long click button or element of device.

Example: || Long Click Element | className=sample class

With Device/ Pararel : || @ {emulator} = | 192.168.1.1 | 192.168.1.2 || Long Click Element | class-Name=sample class | `device_parallel=@ {emulator}`

class mykonos.keywords.element.**ExpectedConditions**

check_element_non_visible (*device=None, *argument, **settings*)

Check element non visible.

The keyword is used to check element non visible.

Example:

|| Check Element Non Visible | className=sampleclassName

With Device/ Pararel : || @ {emulator} = | 192.168.1.1 | 192.168.1.2 || Check Element Non Visible | `device_parallel=@ {emulator}`

Return:

True or False

check_element_visible (*device=None, *argument, **settings*)

Check element visible.

The keyword is used to check element visible.

Example:

|| Check Element Visible | className=sampleclassName With Device/ Pararel : || @ {emulator} = | 192.168.1.1 | 192.168.1.2 || Check Element Visible | `device_parallel=@ {emulator}`

Return:

True or False

page_should_contain_element (*device=None, *argument, **settings*)

Page should contain element. The keyword is used to verify the page is contains locator element.

Example:

|| Page Should Contain Element | className=sample class

With Device/ Pararel : || @ {emulator} = | 192.168.1.1 | 192.168.1.2 || Page Should Contain Element | `device_parallel=@ {emulator}`

Return:

True or False

page_should_contain_text (*device=None, *argument, **settings*)

Page should contain text. The keyword is used to verify the page is contains text.

Example:

|| Page Should Contain Text | text=sample text

With Device/ Pararel : || @ {emulator} = | 192.168.1.1 | 192.168.1.2 || Page Should Contain Text |device_parallel=@ {emulator}

Return:

True or False

page_should_not_contain_element (*device=None, *argument, **settings*)

Page should not contain element.

The keyword is used to verify the page is not contains element.

Example:

|| Page Should Not Contain Element | className=sample element

With Device/ Pararel : || @ {emulator} = | 192.168.1.1 | 192.168.1.2 || Page Should Not Contain Element |device_parallel=@ {emulator}

Return:

True or False

page_should_not_contain_text (*device=None, *argument, **settings*)

Page should not contain text.

The keyword is used to verify the page is not contains text.

Example:

|| Page Should Contain Text | text=sample text

With Device/ Pararel : || @ {emulator} = | 192.168.1.1 | 192.168.1.2 || Page Should Contain Text |device_parallel=@ {emulator}

Return:

True or False

text_should_be_disabled (*device=None, *argument, **settings*)

Text should be disabled.

The keyword is used to identify text disabled.

Example:

|| Element Should Be Disabled | text=sample text

With Device/ Pararel : || @ {emulator} = | 192.168.1.1 | 192.168.1.2 || Element Should Be Disabled |device_parallel=@ {emulator}

Return:

True or False

text_should_be_enabled (*device=None, *argument, **settings*)

Text should be enabled.

The keyword is used to identify text enable.

Example:

|| Text Should Be Enabled | text=sample text

With Device/ Pararel : || @ {emulator} = | 192.168.1.1 | 192.168.1.2 || Text Should Be Enabled |device_parallel=@ {emulator}

Return:

True or False

class mykonos.keywords.element.GetConditions

get_element (*device=None, *argument, **settings*)

Get element info of device . This keyword is used to get element info of device.

Example:

|| Get Element ||

With Device/ Pararel : || @{emulator} = || 192.168.1.1 | 192.168.1.2 || Get Element || **device_parallel=@{emulator}**

Return:

{'currentPackageName': 'com.google.android.apps.nexuslauncher', 'displayHeight': 1794, 'displayRotation': 0, 'displaySizeDpX': 411, 'displaySizeDpY': 731, 'displayWidth': 1080, 'productName': 'sdk_google_phone_x86', 'screenOn': True, 'sdkInt': 25, 'naturalOrientation': True}

get_element_attribute (*device=None, *argument, **settings*)

Get element attribute keyword of device.

List of Elements:

childCount, bounds, className, contentDescription, packageName, resourceName, text, visibleBounds, checkable, checked, clickable, enabled, focusable, disable, focused, longClickable, scrollable, selected

Example:

|| Get Element Attribute | className=sample | element=text

With Device/ Pararel : || @{emulator} = | 192.168.1.1 | 192.168.1.2 || Get Element Attribute | className=sample class | **device_parallel=@{emulator}**

Return:

Attribute from element device

get_element_by_coordinate_x (*device=None, *argument, **settings*)

Get element by coordinate X. This keyword is used to get coordinate X of device.

Example:

|| Get Element By Coordinate X | className=sample class

With Device/ Pararel :

|| @{emulator} = || 192.168.1.1 | 192.168.1.2 || Get Element By Coordinate X || **device_parallel=@{emulator}** || className=sample class

Return:

Coordinate x(int)

get_element_by_coordinate_y (*device=None, *argument, **settings*)

Get element by coordinate Y.

This keyword is used to get coordinate Y of device.

Example:

|| Get Element By Coordinate Y | className=sample class

With Device/ Pararel : || @{emulator} = || 192.168.1.1 | 192.168.1.2 || Get Element By Coordinate Y ||
device_parallel=@{emulator} || className=sample class

Return:

Coordinate y(int)

get_height (device=None)

Get height from display of device.

This keyword is used to get width of device.

Example:

|| Get Height

With Device/ Pararel : || @{emulator} = || 192.168.1.1 | 192.168.1.2 || Get Height || de-
vice_parallel=@{emulator}

Return:

Height of device(int)

get_info (device=None, value=None)

Get Info of Device.

Example:

|| Get Info | value=displayRotation

With Device/ Pararel : || @{emulator} = | 192.168.1.1 | 192.168.1.2 || Get Info |
value=displayRotation | device_parallel=@{emulator}

Return: { u'displayRotation': 0,

u'displaySizeDpY': 640, u'displaySizeDpX': 360, u'currentPackageName':
u'com.android.launcher', u'productName': u'takju', u'displayWidth': 720, u'sdkInt': 18,
u'displayHeight': 1184, u'naturalOrientation': True

}

get_position (position=0, device=None, *argument, **settings)

Get Position of element.

This keyword is used to get position of device element.

Example:

|| Get Position || className=sample || position=1

With Device/ Pararel : || @{emulator} = || 192.168.1.1 || 192.168.1.2 || Get Position || de-
vice_parallel=@{emulator} || className=sample || position=1

Return:

Width of device(int)

get_text (device=None, *argument, **settings)

Get text from element base on locator.

Example:

|| Get Text | className=sample class

With Device/ Pararel : || @ {emulator} = | 192.168.1.1 | 192.168.1.2 || Get Text | class-
Name=sample class | device_parallel=@ {emulator}

Return: String

get_width (*device=None*)

Get width from display of device.

This keyword is used to get width of device,

Example:

|| Get Width

With Device/ Pararel : || @ {emulator} = || 192.168.1.1 | 192.168.1.2 || Get Width || de-
vice_parallel=@ {emulator}

Return:

Width of device(int)

class mykonos.keywords.element.**GlobalElement**

capture_screen (*device=None, location=None*)

Capture screen of device testing, the file name will get automatically by the test case name.

Example:

|| Capture Screen

With file name:

|| Capture Screen | file=sample

With Device/ Pararel : || @ {emulator} = | 192.168.1.1 | 192.168.1.2 || Capture Screen | de-
vice_parallel=\${emulator} || Capture Screen | location=path | device_parallel=@ {emulator}

Return:

screen capture of device(*.png)

clear_text (*device=None, *argument, **settings*)

Clear text on the text field base on locator.

This keywords is used to clear text field.

Example:

||Clear Text | className=sample class

With Device/ Pararel : || @ {emulator} = | 192.168.1.1 | 192.168.1.2 || Clear Text | className=sample class
|device_parallel= @ {emulator}

count_elements (*device=None, *argument, **settings*)

Count total element from the page.

This keywords is used to count total element on the device page.

Example:

|| Count Elements | className=sample class

With Device/ Pararel : || @ {emulator} = | 192.168.1.1 | 192.168.1.2 || Count Element | de-
vice_parallel=@ {emulator}

Return:

Total of elements (int)

dump_xml (*device=None, **settings*)

Dump hierarchy of ui and will be saved as hierarchy.xml.

Example:

|| Dump Xml | file=sample.xml

With Device /pararel :

|| @ {emulator} = | 192.168.1.1 | 192.168.1.2 || Dump Xml | file=sample.xml | de-
vices_parallel=@ {emulator}

Return:

xml file of device

input_text (*device=None, *argument, **settings*)

Input text on the text field base on locator.

This keywords is used to input text into text field.

Example:

|| Input Text | className=sample class input=text

With Device/ Pararel : || @ {emulator} = | 192.168.1.1 | 192.168.1.2 || Input Text | className=sample class
| input=text | device_parallel=@ {emulator}

open_notification (*device=None, **settings*)

Open notification a device.

This keywords is used to open notification of device

Example: || Open notification

With Device/ Pararel : || @ {emulator} = | 192.168.1.1 | 192.168.1.2 || Open notification | de-
vices_parallel=@ {emulator}

open_quick_settings (*device=None, **settings*)

Open Quick Setting a device.

This keywords is used to open setting of device

Example:

|| Open Quick setting |

With Device/ Pararel : || @ {emulator} = | 192.168.1.1 | 192.168.1.2 || Open Quick setting | device_parallel=
@ {emulator}

turn_off_screen (*device=None, **settings*)

Turn off Screen Device.

Example:

|| Turn Off Screen

With Device/ Pararel : || @ {emulator} = | 192.168.1.1 | 192.168.1.2 || Turn Off Screen | de-
vice_parallel=@ {emulator}

Return:

True or False

turn_on_screen (*device=None, **settings*)

Turn on Screen Device.

Example:

|| Turn On Screen

With Device/ Pararel : || @{{emulator}} = | 192.168.1.1 | 192.168.1.2 || Turn On Screen | device_parallel=@{{emulator}}

Return:

True or False

class mykonos.locator.locator_element.**LocatorElement**

down_position (*parent, *argument, **settings*)

Access left position from device.

Example:

|| \${{locator}}= Get Locator | text=sample text

|| \${{down}}= Down Position | parent=\${{locator}} text=sample text

get_child (*parent, *argument, **settings*)

Access child locator from device.

Example:

|| \${{locator}}= Get Locator | text=sample text

|| \${{child}}= Get Child | parent=\${{locator}} text=sample text

get_locator (*device=None, *argument, **settings*)

Access locator from device.

selector support:

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

Example:

|| \${{locator}}= Get Locator | text=sample text

With Device: || @{{emulator}} = | 192.168.1.1 | 192.168.1.2 || \${{locator}}= Get Locator | text=sample text | devices_parallel=@{{emulator}}

get_locator_by_index (*device=None, *argument, **settings*)

Get Element locator by index on device.

Example:

|| Get Locator By Index | text=sample_text | index=1

```

    || ${locator}= Get Locator | text=sample text
    || Get Locator By Index | locator=${locator} | index=1
get_sibling (parent, *argument, **settings)
    Access sibling locator from device.

    Example:

    || ${locator}= Get Locator | text=sample text
    || ${sibling}= Get Sibling | parent=${locator} text=sample text

handlers (action, function)
    Call customized function on device.

    Example:

    || Handlers | action=on | function=sample_function

left_position (parent, *argument, **settings)
    Access left position from device.

    Example:

    || ${locator}= Get Locator | text=sample text
    || ${left}= Left Position | parent=${locator} text=sample text

right_position (parent, *argument, **settings)
    Access left position from device.

    Example:

    || ${locator}= Get Locator | text=sample text
    || ${right}= Right Position | parent=${locator} text=sample text

up_position (parent, *argument, **settings)
    Access left position from device.

    Example:

    || ${locator}= Get Locator | text=sample text
    || ${up}= Up Position | parent=${locator} text=sample text

class mykonos.locator.locator_element.WatcherElement
    Class is used to perform some actions when selector cannot be found.

watcher (**settings)
    Watcher is registered when a selector cannot be find.

    name=name of watcher
    WHEN, className=sample_class
    WHEN, packageName=sample_package

    Example:

    || ${sample_watcher}=name=sample_watcher | className=sample_class
    || Click Element | watcher=${sample_watcher} | text=sample_text

watcher_action (action, **settings)
    Watcher Action is used to running the action on the watcher.

    run=Force to run all watchers

```

remove=Remove watchers
reset=Reset all triggered watchers
list=List all watchers
triggered=Check if there is any watcher triggered

Example:

```
// Watcher Action | action=run
// Watcher Action | action=remove
// Watcher Action | action=remove | name=sample_watcher
// Watcher Action | action=reset
// Watcher Action | action=list
// Watcher Action | action=triggered
```

class mykonos.keywords.management_device.**ManagementDevice**

close_all_app (*devices_parallel*)

Close all tasks on device, and kill all application sessions. **Example:** `// Close All App | devices_parallel=emulator-554 |`

close_app (*device, package*)

Close Application the device. This keywords is used to close the current application and kill session on device. **Example:** `// Close App | devices_parallel=${emulator} | package=Package Activity`

hide_keyboard ()

Hide Keyword on Device. This keyword is used to hide keyboard device. **Example:** `// Hide keyboard |`

open_app (***settings*)

Open Application on device. This keyword is used to open new applications. **Example:** `// Open Application | devices_parallel=emulator-554 | package=sample_apk`

pull (***settings*)

Pull file from Device. This Keyword is used to retrieves file from device. **Example:** `// Pull | local=sample_path | remote=sample_location // Pull | local=sample_path |`

push (***settings*)

Push file into Device. This keyword is used to put file in specific path of device. **Example:** `// Push | local=sample_path | remote=sample_location`

quit_app (*devices_parallel, package*)

Quit application on device. This keyword is used to close application without kill a session. **Example:** `// Quit App | package=sample_apk`

reset_app (*devices_parallel, package*)

Reset Application on Device. This keyword is used to reset the current application while session is keep alive. **Example:** `// Reset Application | emulator=emulator-554 | package=sample_apk`

scan_current_device (**args, **settings*)

Scan current device on the workstation, and consume to open application. **Example:** `// Scan Current Device | emulator-554`

switch_application (*devices_parallel, new_app*)

Switch application the devices. This keywords return previous active application and it can be used in the next application. **Example:** `// Switch Application | devices_parallel=sample_device | new_app=sample_app`

```
class mykonos.keywords.key_event.KeyEvent
```

```
long_press (*args, **setting)
```

Long press on device.

This keyword is used to press key on device with optional duration.

Example:

```
|| Long Press |back |timer=100
```

Return: True or False

```
press_keycode (device=None, *argument, **settings)
```

Press key on device.

This keyword is used to press key on device.

Key Support:

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

Example:

```
|| Press Keycode |keys=back
```

```
class mykonos.keywords.touch.Touch
```

```
drag_screen (sx, sy, ex, ey, steps, device=None)
```

Geasture drag interanction on device. This keyword is used to drag another point ui object to another point ui object. **Example:** || Drag Screen | sx=189 | sy=210 | ex=954 | ey=336 | steps=100

```
fling (*argument, **settings)
```

Fling interanction on Android device. This keyword is used to perform fling to spesific ui object. **Example:** How to user without action: || Fling How to use fling horizontal: || Fling | action=horizontal forward || Fling | max_swipes=1 action=horizontal to begining || Fling | action=horizontal backward || Fling | action=horizontal to end How to use scroll vertical: || Fling | action=vertical forward || Fling | max_swipes=1 action=vertical to begining || Fling | action=vertical backward || Fling | action=vertical to end

```
pinch (device=None, *argument, **settings)
```

Pinch interaction on Device **Example:** || Pinch | steps=100 action=In percent=100 || Pinch | steps=100 action=Out percent=100

```
scroll (device=None, *argument, **settings)
```

Scroll interanction on device. This keyword is used to perfom scroll on device. **Example:** How to use scroll horizontal: || Scroll | steps=100 || Scroll | steps=100 action=horizontal forward || Scroll | steps=100 max_swipes=1 action=horizontal to begining || Scroll | textName='Calculator' className='sampleClass' action=horizontal to || Scroll | action=horizontal backward || Scroll | action=horizontal to end How to use scroll vertical: || Scroll | steps=100 || Scroll | steps=100 action=vertical forward || Scroll | steps=100 max_swipes=1 action=vertical to begining || Scroll | textName='Calculator' className='sampleClass' action=vertical to || Scroll | action=vertical backward || Scroll | action=vertical to end

swipe (*sx, sy, ex, ey, steps=10, device=None, **settings*)

Geasture swipe with interanction on device. Swipe from (sx, sy) to (ex, ey). **Example:** ||Swipe | sx=10 sy=10 ex=20 ey=20 | steps=100

swipe_with_direction (*device=None, *argument, **settings*)

Gesture swipe with direction on device. Swipe with direction : right, left, up and down **Example:** ||Swipe | direction=right | steps=100 ||Swipe | direction=left | steps=100 ||Swipe | direction=up | steps=100 ||Swipe | direction=down | steps=100

class mykonos.keywords.wait.**Wait**

wait_until_element_is_exists (*time=1000, error=None, *argument, **settings*)

This keyword is used to wait until spesific element is exists.

Example:

|| Wait Until Element Is Exists | className=sample class or text=Sample Text | erro=sample error

wait_until_page_contains (*device=None, error=None, time=1000, *argument, **settings*)

This keyword is used to wait until page is contain spesific element.

Example:

|| Wait Until Page Contains | className= sample class | error=sample error

wait_until_page_does_not_contains (*time=1000, *argument, **settings*)

This keyword is used to wait until page is not contain spesific element.

Example:

|| Wait Until Page Does Not Contains| className=sample class

CHAPTER 5

Indices and tables

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

m

- `mykonos.keywords.element`, [7](#)
- `mykonos.keywords.key_event`, [16](#)
- `mykonos.keywords.management_device`, [16](#)
- `mykonos.keywords.touch`, [17](#)
- `mykonos.keywords.wait`, [18](#)
- `mykonos.locator.locator_element`, [14](#)

C

capture_screen() (*mykonos.keywords.element.GlobalElement*
method), 12

check_element_non_visible()
(*mykonos.keywords.element.ExpectedConditions*
method), 8

check_element_visible()
(*mykonos.keywords.element.ExpectedConditions*
method), 8

clear_text() (*mykonos.keywords.element.GlobalElement*
method), 12

Click (class in *mykonos.keywords.element*), 7

click_a_point() (*mykonos.keywords.element.Click*
method), 7

click_element() (*mykonos.keywords.element.Click*
method), 7

close_all_app() (*mykonos.keywords.management_device.ManagementDevice*
method), 16

close_app() (*mykonos.keywords.management_device.ManagementDevice*
method), 16

count_elements() (*mykonos.keywords.element.GlobalElement*
method), 12

D

down_position() (*mykonos.locator.locator_element.LocatorElement*
method), 14

drag_screen() (*mykonos.keywords.touch.Touch*
method), 17

dump_xml() (*mykonos.keywords.element.GlobalElement*
method), 13

E

ExpectedConditions (class in
mykonos.keywords.element), 8

F

fling() (*mykonos.keywords.touch.Touch* method), 17

G

getChild() (*mykonos.locator.locator_element.LocatorElement*
method), 14

get_element() (*mykonos.keywords.element.GetConditions*
method), 10

get_element_attribute()
(*mykonos.keywords.element.GetConditions*
method), 10

get_element_by_coordinate_x()
(*mykonos.keywords.element.GetConditions*
method), 10

get_element_by_coordinate_y()
(*mykonos.keywords.element.GetConditions*
method), 10

get_height() (*mykonos.keywords.element.GetConditions*
method), 11

get_id() (*mykonos.keywords.element.GetConditions*
method), 11

get_locator() (*mykonos.locator.locator_element.LocatorElement*
method), 14

get_locator_by_index()
(*mykonos.locator.locator_element.LocatorElement*
method), 14

get_position() (*mykonos.keywords.element.GetConditions*
method), 11

get_sibling() (*mykonos.locator.locator_element.LocatorElement*
method), 15

get_text() (*mykonos.keywords.element.GetConditions*
method), 11

get_width() (*mykonos.keywords.element.GetConditions*
method), 12

GetConditions (class in *mykonos.keywords.element*),
10

GlobalElement (class in *mykonos.keywords.element*),
12

H

handlers() (*mykonos.locator.locator_element.LocatorElement*
method), 15

`hide_keyboard()` (*mykonos.keywords.management_device.ManagementDevice* *method*), 16
I
`input_text()` (*mykonos.keywords.element.GlobalElement* *method*), 13
K
`KeyEvent` (*class in mykonos.keywords.key_event*), 16
L
`left_position()` (*mykonos.locator.locator_element.LocatorElement* *method*), 15
`LocatorElement` (*class in mykonos.locator.locator_element*), 14
`long_click_element()` (*mykonos.keywords.element.Click* *method*), 8
`long_press()` (*mykonos.keywords.key_event.KeyEvent* *method*), 17
M
`ManagementDevice` (*class in mykonos.keywords.management_device*), 16
`mykonos.keywords.element` (*module*), 7
`mykonos.keywords.key_event` (*module*), 16
`mykonos.keywords.management_device` (*module*), 16
`mykonos.keywords.touch` (*module*), 17
`mykonos.keywords.wait` (*module*), 18
`mykonos.locator.locator_element` (*module*), 14
O
`open_app()` (*mykonos.keywords.management_device.ManagementDevice* *method*), 16
`open_notification()` (*mykonos.keywords.element.GlobalElement* *method*), 13
`open_quick_settings()` (*mykonos.keywords.element.GlobalElement* *method*), 13
P
`page_should_contain_element()` (*mykonos.keywords.element.ExpectedConditions* *method*), 8
`page_should_contain_text()` (*mykonos.keywords.element.ExpectedConditions* *method*), 8
`page_should_not_contain_element()` (*mykonos.keywords.element.ExpectedConditions* *method*), 9
`page_should_not_contain_text()` (*mykonos.keywords.element.ExpectedConditions* *method*), 9
`pinch()` (*mykonos.keywords.touch.Touch* *method*), 17
`press_keycode()` (*mykonos.keywords.key_event.KeyEvent* *method*), 17
`pull()` (*mykonos.keywords.management_device.ManagementDevice* *method*), 16
`push()` (*mykonos.keywords.management_device.ManagementDevice* *method*), 16
Q
`quit_app()` (*mykonos.keywords.management_device.ManagementDevice* *method*), 16
R
`reset_app()` (*mykonos.keywords.management_device.ManagementDevice* *method*), 16
`right_position()` (*mykonos.locator.locator_element.LocatorElement* *method*), 15
S
`scan_current_device()` (*mykonos.keywords.management_device.ManagementDevice* *method*), 16
`scroll()` (*mykonos.keywords.touch.Touch* *method*), 17
`swipe()` (*mykonos.keywords.touch.Touch* *method*), 17
`swipe_with_direction()` (*mykonos.keywords.touch.Touch* *method*), 18
`switch_application()` (*mykonos.keywords.management_device.ManagementDevice* *method*), 16
T
`toggle_notification_be_disabled()` (*mykonos.keywords.element.ExpectedConditions* *method*), 9
`text_should_be_enabled()` (*mykonos.keywords.element.ExpectedConditions* *method*), 9
`Touch` (*class in mykonos.keywords.touch*), 17
`turn_off_screen()` (*mykonos.keywords.element.GlobalElement* *method*), 13
`turn_on_screen()` (*mykonos.keywords.element.GlobalElement* *method*), 13
U
`up_position()` (*mykonos.locator.locator_element.LocatorElement* *method*), 15
W
`Wait` (*class in mykonos.keywords.wait*), 18

`wait_until_element_is_exists()`
 (*mykonos.keywords.wait.Wait method*), 18

`wait_until_page_contains()`
 (*mykonos.keywords.wait.Wait method*), 18

`wait_until_page_does_not_contains()`
 (*mykonos.keywords.wait.Wait method*), 18

`watcher()` (*mykonos.locator.locator_element.WatcherElement*
 method), 15

`watcher_action()` (*mykonos.locator.locator_element.WatcherElement*
 method), 15

`WatcherElement` (class in
 mykonos.locator.locator_element), 15