# Poco使用手册

## Poco简介

Poco是一个强大的类库，并易于构建自己的应用程序。

Poco帮助你创建跨平台的应用程序（一次编写，多平台编译和运行）。

模块化和可扩展，可应用于嵌入式到企业级程序。

提供全面，易懂的编程接口。

使用C++语言，快速且高效。

Poco崇尚简易。

在设计，代码风格和文档上保持一致。

Poco强调代码质量，包括可读性，综合性，一致性，编码风格和可测试性。

Poco使得C++编程更加容易。

## 安装

pip install pocoui

（截止到2019年4月，poco的最新版本为1.0.76）

## 调用

### Unity

from poco.drivers.unity3d import UnityPoco

poco = UnityPoco()

# for unity editor on windows

# poco = UnityPoco(('localhost', 5001), unity\_editor=True)

ui = poco('...')

ui.click()

### Android native APP

from poco.drivers.android.uiautomation import AndroidUiautomationPoco

poco = AndroidUiautomationPoco(use\_airtest\_input=True, screenshot\_each\_action=False)

### Cocos-lua

from poco.drivers.std import StdPoco

poco = StdPoco()

### Cocos-js

from poco.drivers.cocosjs import CocosJsPoco

poco = CocosJsPoco()

### iOS

from poco.drivers.ios import iosPoco

poco = iosPoco()

### Std-broker

from poco.drivers.std import StdPoco

from poco.utils.device import VirtualDevice

poco = StdPoco(15004, VirtualDevice('localhost'))

## Poco类

**classPoco(agent, \*\*options)**

是poco.acceleration.PocoAccelerationMixin的子类

### 参数：

#### action\_interval:

time interval to wait for the action (such as touch or swipe) completion performed on device and for the UI to become still (stable). Default value is 0.8s.

等待在设备上执行的动作(如触摸或滑动)完成和UI变得静止(稳定)的时间间隔。默认值为0.8s。

#### poll\_interval:

the minimum time needed between each poll events (such as waiting for UI element to appear on the screen). Polling is done periodically.

每个轮询事件之间所需的最短时间(例如等待UI元素出现在屏幕上)。轮询是定期进行的。

#### pre\_action\_wait\_for\_appearance:

time interval to wait before the action (such as click or swipe) is performed. If the target UI element does not appear on the screen after this time interval, the [PocoNoSuchNodeException](https://poco-chinese.readthedocs.io/zh_CN/latest/source/poco.exceptions.html#poco.exceptions.PocoNoSuchNodeException) is raised

在执行操作(例如单击或滑动)之前等待的时间间隔。如果目标UI元素在此时间间隔后不出现在屏幕上，则会引发PocoNoSuchNodeException的异常。

#### touch\_down\_duration:

Touch down step duration of the click operation last for. If this argument is provided, this value will set to self.agent.input module. Note that not all implementation of poco support this parameter. If not support, you may see a warning.

按下持续的单击操作的步骤持续时间。如果提供了此参数，则此值将设置到self.agent.input模块。注意，并非所有poco实现都支持此参数。如果不支持，您可能会看到警告。

### 方法

#### \_\_init\_\_

初始化方法，定义Poco类的各项参数。

**def** \_\_init\_\_(self, agent, \*\*options):

super(Poco, self).\_\_init\_\_()

self.\_agent = agent

*# options*

self.\_pre\_action\_wait\_for\_appearance = options.get('pre\_action\_wait\_for\_appearance', 6)

self.\_post\_action\_interval = options.get('action\_interval', 0.8)

self.\_poll\_interval = options.get('poll\_interval', 1.44)

**if** 'touch\_down\_duration' **in** options:

touch\_down\_duration = options['touch\_down\_duration']

**try**:

touch\_down\_duration = float(touch\_down\_duration)

**except** ValueError:

**raise** ValueError('Option `touch\_down\_duration` should be <float>. Got *{}*'

.format(repr(touch\_down\_duration)))

self.\_agent.input.setTouchDownDuration(touch\_down\_duration)

self.\_pre\_action\_callbacks = [self.\_\_class\_\_.on\_pre\_action]

self.\_post\_action\_callbacks = [self.\_\_class\_\_.on\_post\_action]

self.\_agent.on\_bind\_driver(self)

#### \_\_call\_\_

调用方法，通过查询表达式选择UI元素。

**def** \_\_call\_\_(self, name=**None**, \*\*kw):

*"""*

*Call Poco instance to select the UI element by query expression. Query expression can contain specific name*

*and/or other attributes. Invisible UI elements will be skipped even if "visible=False" argument is set.*

*Selection process is not executed instantly, the query expression is stored in the UI proxy and the selection is*

*executed only then when the UI element(s) info is required (such get the point coordinates where to click,*

*and/or retrieve the specific attribute value).*

*Examples:*

*This example shows selecting a Button named 'close'::*

*poco = Poco(...)*

*close\_btn = poco('close', type='Button')*

*Args:*

*name (:obj:`str`): name of the UI element to be selected*

*Keyword Args:*

*xx: arbitrary key value pair that stands for selecting the UI matching the value of ``UI.xx``*

*xxMatches (:obj:`str`): arbitrary key value pair that stands for selecting the UI matching the regular*

*expression pattern ``UI.xx``*

*In keyword args, you can only use `xx` or `xxMatches` at the same time. Using both with the same attribute does*

*not make sense. Besides, `xx` should not start with ``\_`` (underscore) as attributes start with ``\_`` are*

*private attributes that used by sdk implementation.*

*::*

*# select the UI element(s) which text attribute matches the pattern '^close.\*$'*

*poco = Poco(...)*

*arb\_close\_btn = poco(textMatches='^close.\*$')*

*Returns:*

*:py:class:`UIObjectProxy <poco.proxy.UIObjectProxy>`: UI proxy object representing the UI element matches*

*the given query expression.*

*"""*

**if** **not** name **and** len(kw) == 0:

warnings.warn("Wildcard selector may cause performance trouble. Please give at least one condition to "

"shrink range of results")

**return** UIObjectProxy(self, name, \*\*kw)