# 使用说明

*当前项目进度大概22%，但不影响基本的case开发. 后期还会加入各种功能。例如不需要写代码来写脚本。*

**配置环境**

**1. 安装Python 2.7.13**

<https://www.python.org/ftp/python/2.7.13/python-2.7.13.msi>

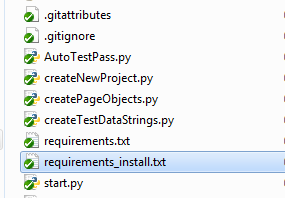
**2. 更新pip**

在控制台运行

python -m pip install --upgrade pip

**3. 解压项目**

例如解压到D:盘根目录

****

**3. 安装requirement\_install.txt**

pip install -r <PATH>\AutoTestPass\requirements\_install.txt

例如

pip install -r D:\AutoTestPass\requirements\_install.txt

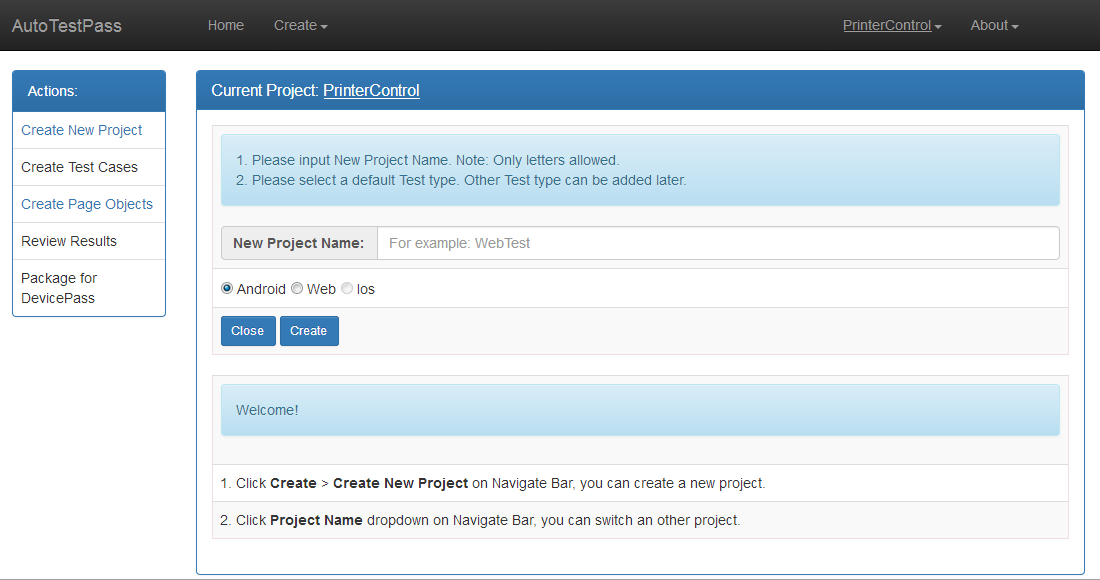
**项目使用**

**1. 打开UI (需要IE11, Firefox 或者Chrome)**

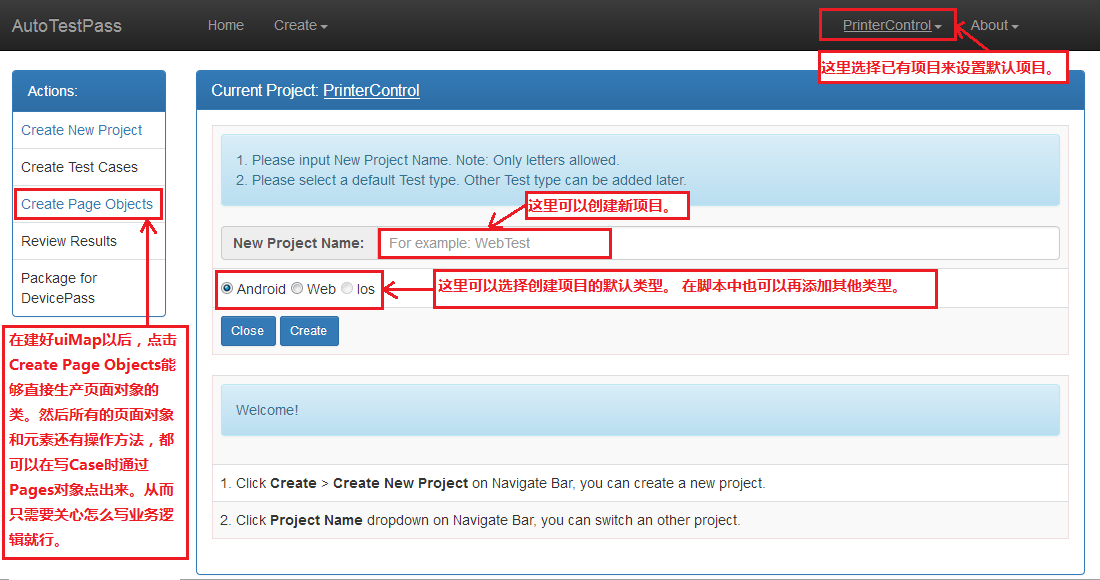
在控制台或IDE中运行

python <PATH>AutoTestPass.py

UI将会自动打开。如下图:



**1.1 UI重要功能说明**



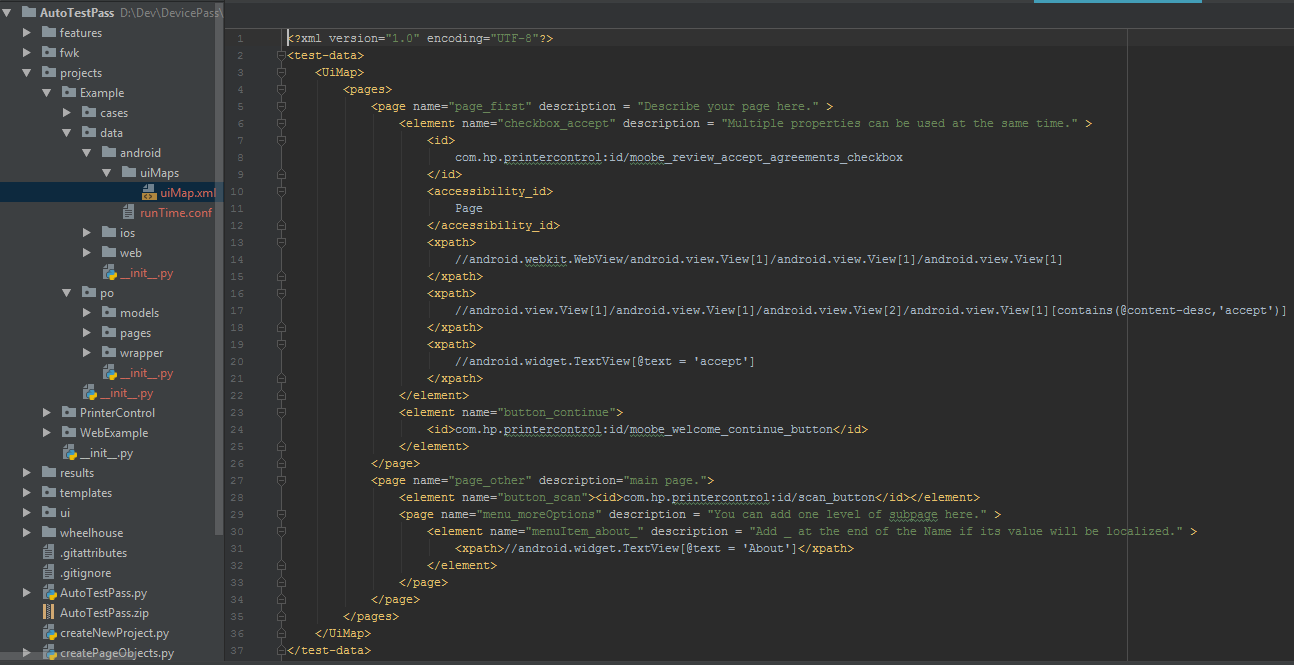
**1.2 建立新项目**

以建立Example项目为例。建好后得到如下提示步骤。

1. The Project **Example** is created successfully.  
2. Please fill out **D:\Dev\DevicePass\script\AutoTestPass\projects\Example\data\android or web or ios\runTime.conf**.  
3. Please fill out **D:\Dev\DevicePass\script\AutoTestPass\projects\Example\data\android or web or ios\uiMaps\uiMap.py** and then click **Create Page Objects** to create page objects.  
4. Please write Test Suite(cases) in **D:\Dev\DevicePass\script\AutoTestPass\projects\Example\cases\Example.py**.  
5. Please add the names of Test cases that you want to test from the Test Suite to **D:\Dev\DevicePass\script\AutoTestPass\(start\_)Example.py**.

**1.3. uiMap.xml**

**这个文件存放所有元素的locater**

****

<page name="page\_first" description = "Describe your page here." >  
 <element name="checkbox\_accept" description = "Multiple properties can be used at the same time." >  
 <id>  
 com.hp.printercontrol:id/moobe\_review\_accept\_agreements\_checkbox  
 </id>  
 <accessibility\_id>  
 Page  
 </accessibility\_id>  
 <xpath> //android.webkit.WebView/android.view.View[1]/android.view.View[1]/android.view.View[1]  
 </xpath>  
 <xpath> //android.view.View[1]/android.view.View[1]/android.view.View[2]/android.view.View[1][contains(@content-desc,'accept')]  
 </xpath>  
 <xpath>  
 //android.widget.TextView[@text = 'accept']  
 </xpath>  
 </element>  
 <element name="button\_continue">  
 <id>com.hp.printercontrol:id/moobe\_welcome\_continue\_button</id>  
 </element>  
 </page>

一个element元素可以写多个locator, 甚至可以写多个相同的locator标签。例如，

<id>xxx1</id>

<id>xxx2</id>

<xpath>yyy1</xpath>

<xpath>yyy2</xpath>

其中任何一个能被脚本找到，脚本就会用哪个。

<page name="page\_other" description="main page.">  
 <element name="button\_scan">

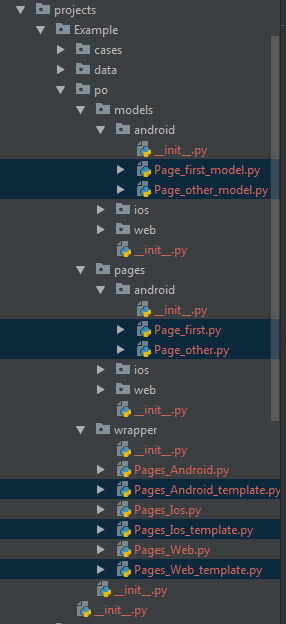
<id>com.hp.printercontrol:id/scan\_button</id></element>  
 <**page** name="menu\_moreOptions" description = "You can add one level of subpage here." >  
 <element name="menuItem\_about\_" description = "Add \_ at the end of the Name if its value will be localized." >  
 <xpath>//android.widget.TextView[@text = 'About']</xpath>  
 </element>  
 </**page**>  
</page>

每个Page对象可以有且仅有一级子Page

**当修改成你自己的测试对象后，可以点击Create Page Objects去生成页面对象。如需要修改uiMap文件，可以点击Create Page Objects再次生成页面对象。**

**1.3. 生成的页面对象文件**

在运行了Create Page Objects后，会自动生成类似以下被选择的文件.

****

**Example > po > models > \*\_model.py**

保存了当前Page所有对象，不添加删除元素，不需要修改。(重新运行Create Page Objects后，此类文件会被重写)

**import** inspect  
**from** fwk.page.AndroidCommonPage **import** AndroidCommonPage  
  
  
'''Describe your page here.'''  
**class Page\_first\_model**(AndroidCommonPage):  
 page\_name = 'page\_first'  
  
 **def** \_\_init\_\_(self):  
 self.\_\_AndroidCommonPage = AndroidCommonPage.\_\_init\_\_(self)  
  
 **def checkbox\_accept**(self):  
 **return** self.get(inspect.stack()[0][3])  
  
 **def button\_continue**(self):  
 **return** self.get(inspect.stack()[0][3])

**Example > po > pages >\*.py**

这是页面类，可以在里面写一些项目通用的逻辑。(重新运行Create Page Objects后，此类文件不会被重写)

# coding: utf-8  
**from** projects.Example.po.models.android.Page\_first\_model **import** Page\_first\_model  
  
  
'''Describe your page here.'''  
**class Page\_first**(Page\_first\_model):  
 **def** \_\_init\_\_(self, UI):  
 self.UI = UI  
 Page\_first\_model.\_\_init\_\_(self)  
  
 # This is function template of how to write your Buissness Logic.  
 **def example**(self):  
 **pass** # self.checkbox\_accept().waitForShown().click()  
 # self.Pages.Page\_endUserLicenseAgreement.text\_always().clickIfPresent().wait(1)  
 # self.button\_continue().click()  
 # self.button\_search().waitForShown(60).click().wait(1)  
 # self.Edit\_search().setValue("10.10.63.128")  
 # self.text\_printerIp().relocateByText("10.10.63.128").click()  
 # self.image\_appIcon().verifyIsShown()  
 # A few elements' properties will be changed after finding. It should be found again by using clearForRefinding()  
 # self.image\_appIcon().waitForShown().clearForRefinding().click()  
 # self.text\_version().verifyEqual(self.text\_version().getValue(), "4.3.19")

**1.4写cases**

**在下面文件里面写逻辑**

**Example > cases > Example.py**

@classmethod  
**def setUpClass**(cls):  
 cls.setUpBeforClass() #setup test before starting.  
 cls.Pages\_Android = Pages\_Android(cls.UI\_Android) # create page objects of Android test.  
 # cls.Pages\_Web = Pages\_Web(cls.UI\_Web)  
 # cls.Pages\_Ios = Pages\_Android(cls.UI\_Ios)  
 cls.Pages = cls.Pages\_Android # Just make it simple, you can ignore this step.

在这个**保留方法**里面得到你想用的Pages对象。Pages\_Android, Pages\_Web and Pages\_Ios可以单独或同时使用.

所有元素和对象都可以直接点出来。所以现在开发用力只需要关系测试逻辑。

self.Pages.Page\_first.button\_continue().waitForShown()  
self.Pages.Page\_first.button\_continue().click()  
self.Pages.Page\_first.button\_continue().clickIfPresent()

**1.5 运行cases**

**Example > start\_Example.py**

如果需要在DevicePass中运行， 去掉\_Example

# coding: utf-8  
**from** projects.Example.cases.Example **import** Example  
  
**import** unittest  
**import** sys  
reload(sys)  
sys.setdefaultencoding('utf-8')  
  
**if** \_\_name\_\_ == '\_\_main\_\_':  
 suite = unittest.TestSuite()  
  
 suite.addTest(Example("test\_your\_flow"))  
 suite.addTest(Example("test\_your\_other\_flow"))  
  
 test\_result = unittest.TextTestRunner(verbosity=2).run(suite)

**其他**

**1. Log**

**可以直接在log中查看结果**

2017-05-09 16:08:49,099 INFO - projects: [NA]

2017-05-09 16:08:49,099 INFO - testName: [HomeMoreAbout]

2017-05-09 16:08:49,099 INFO - os: [Android]

2017-05-09 16:08:49,099 INFO - version: [5.0.2]

2017-05-09 16:08:49,099 INFO - language: [en]

2017-05-09 16:08:49,099 INFO - region: [US]

2017-05-09 16:08:49,099 INFO - deviceName: [HC43YWW01974]

2017-05-09 16:08:49,099 INFO - deviceModel: [HTC D816w]

test\_flow (projects.PrinterControl.cases.HomeMoreAbout.HomeMoreAbout) ... 2017-05-09 16:08:49,599 DEBUG - \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2017-05-09 16:08:49,599 DEBUG - \*\*\*\*\*\*\*\*Start Testcase\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2017-05-09 16:08:49,650 INFO - Waiting for that the element [checkbox\_accept] is shown on page['page\_agreements'].

2017-05-09 16:08:52,190 INFO - ......Finding element [checkbox\_accept] of page [page\_agreements]. 2s elapsed. Timeout is 30.0s. Interval is 2.

2017-05-09 16:08:52,230 INFO - Click element [checkbox\_accept] on [page\_agreements] page.

2017-05-09 16:08:52,940 INFO - Click element [button\_continue] on [page\_agreements] page.

2017-05-09 16:08:53,410 INFO - Waiting for that the element [link\_setupLater] is shown on page['page\_welcome'].

2017-05-09 16:08:54,519 INFO - Click element [link\_setupLater] on [page\_welcome] page.

ok

2017-05-09 16:08:55,880 DEBUG - ------------------------------------------------------------------------------

2017-05-09 16:08:55,880 DEBUG - [[[Step]]]: 1

2017-05-09 16:08:55,880 DEBUG - [[[Description]]]: 1. Install AiO app / clear data of AiO app.~!~2. Launch AiO app.~!~3. Go to Home screen.

2017-05-09 16:08:55,880 DEBUG - [[[ExpectedResult]]]: AiO Home screen is displayed.

2017-05-09 16:08:55,880 DEBUG - [[[ErrorMessage]]]: NA

2017-05-09 16:08:55,880 DEBUG - [[[ManualCheck]]]: #$#Step ended@@@00001\_HomeMoreAbout\1\_stepEnd.png#$#~!~

2017-05-09 16:08:55,880 DEBUG - [[[Result]]]: Pass

2017-05-09 16:08:55,880 DEBUG - [[[Time(s)]]]: 5

2017-05-09 16:08:55,880 DEBUG - \*\*\*\*\*\*\*\*End Testcase\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2017-05-09 16:08:55,880 DEBUG - \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

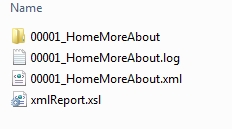
----------------------------------------------------------------------

Ran 1 test in 17.527s

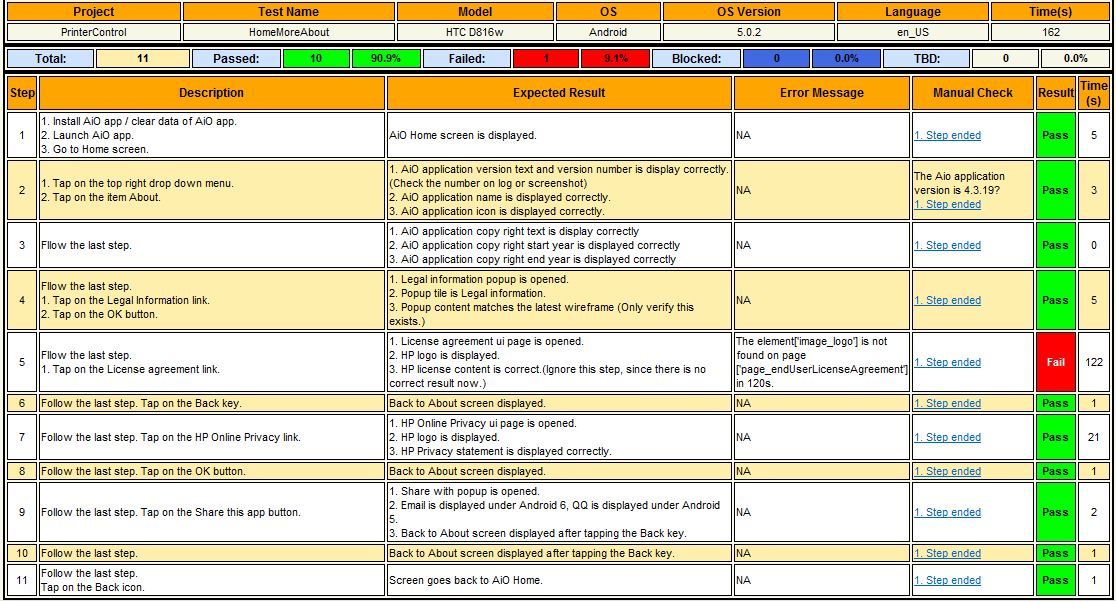
OK

**2. 报告**

每次运行都会生成一个报告，在DevicePass中，也可以下载来看



用浏览器打开.xml来查看



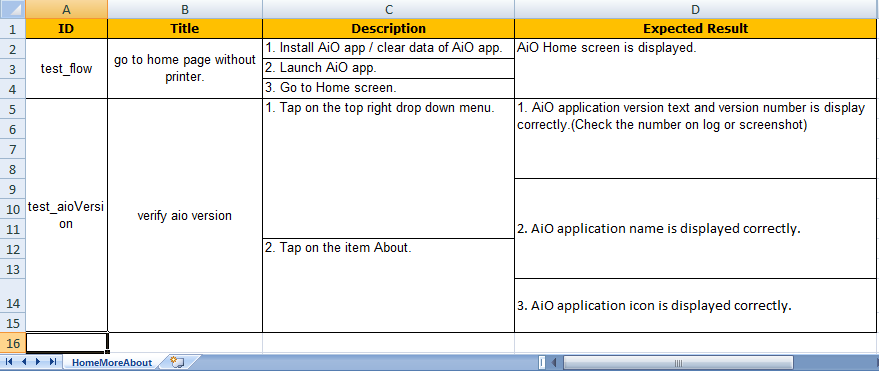
**2.1 Description & Expected Result:**

self.Result.setDescription("1. Install AiO app / clear data of AiO app.",  
 "2. Launch AiO app.",  
 "3. Go to Home screen.")  
self.Result.setExpectedResult("AiO Home screen is displayed.")

这两个方法对应Descripttion和Expected Result.

这2个代码也可以省略直接写在caseInfo.xlsx中,

**def test\_flow**(self):  
 self.Result.setDescriptionAndExpectedResultFromExcel("test\_flow")



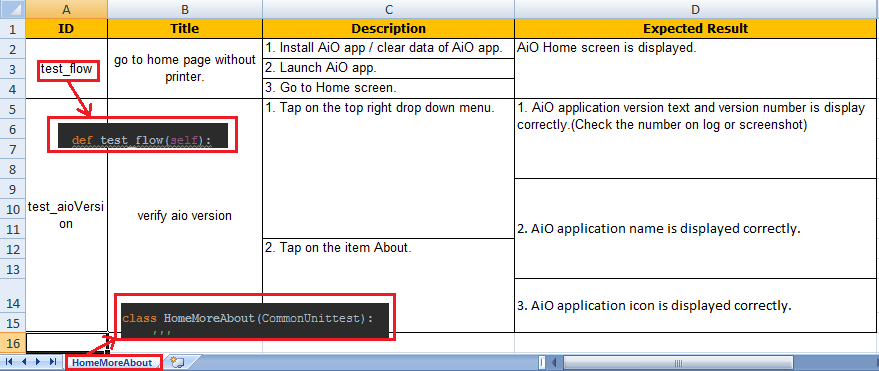
通过下面的方法来调用。

cls.Result.loadAndroidCaseInfoFromExcel()

cls.Result.loadIosCaseInfoFromExcel()

cls.Result.loadWebCaseInfoFromExcel()

如果Sheet Name对应了Class Name, ID对应了def name，程序运行时Description和Expected Result将会自动被调用。



也可以自由决定读哪个sheet的内容:

cls.Result.loadAndroidCaseInfoFromExcel("HomeMoreAbout")

同时也可以自定义读取所需的Description和ExpectedResult:

**2.2 添加screenshot和comment:**

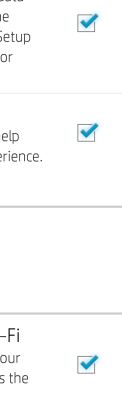
self.Result.addScreenshot("begin", "This is a begin.")

这个方法可以截图。可以截取多次，都会在报告上显示。

self.Result.addComment("It is right.")

这个方法可以加批注。可以加多次，都会在报告上显示。

**2.3 处理同一屏相同属性元素:**

****

例如如图的情况, 除了用xpath可以分别得到对用的3个Checkbox的属性。

AutoTestPass 还提供了以下方法:

**2.3.1 分别设置index**

<element name="checkBox\_usageTracking"><id index="1">android:id/checkbox</id></element>  
<element name="checkBox\_hpSuppliesShopping"><id index="2">android:id/checkbox</id></element>  
<element name="checkBox\_wirelessNetwork"><id index="3">android:id/checkbox</id></element>

在uiMap.xml设置相应的index

调用方法:

self.Pages.Page\_appSettings.checkBox\_usageTracking().click()  
self.Pages.Page\_appSettings.checkBox\_hpSuppliesShopping().click()  
self.Pages.Page\_appSettings.checkBox\_wirelessNetwork().click()

**2.3.2 设置index为0**

<element name="checkBox\_All\_setIndex"><id index="0">android:id/checkbox</id></element>

调用方法:

self.Pages.Page\_appSettings.checkBox\_All\_setIndex().getItem(1).click()  
self.Pages.Page\_appSettings.checkBox\_All\_setIndex().getItem(2).click()  
self.Pages.Page\_appSettings.checkBox\_All\_setIndex().getItem(3).click()

**2.3.3 不设置index**

<element name="checkBox\_All\_noIndex"><id>android:id/checkbox</id></element>

调用方法:

self.Pages.Page\_appSettings.checkBox\_All\_noIndex().getItems().getItem(1).click()  
self.Pages.Page\_appSettings.checkBox\_All\_noIndex().getItems().getItem(2).click()  
self.Pages.Page\_appSettings.checkBox\_All\_noIndex().getItems().getItem(3).click()

**2.4 验证Checked状态:**

self.UI\_Android.verifyChecked(self.Pages.Page\_appSettings.checkBox\_usageTracking().isChecked())

下面任何一种也可以

1. self.UI\_Android.verifyChecked(self.Pages.Page\_appSettings.checkBox\_usageTracking().getValue())  
2. self.Pages.Page\_appSettings.UI.verifyChecked(self.Pages.Page\_appSettings.checkBox\_usageTracking().getValue())  
3. self.UI\_Android.verifyEqual(self.Pages.Page\_appSettings.checkBox\_usageTracking().getValue(), self.UI\_Android.VerifyString.CHECKED)  
4. self.Pages.Page\_appSettings.UI.verifyEqual(self.Pages.Page\_appSettings.checkBox\_usageTracking().getValue(), self.UI\_Android.VerifyString.CHECKED)

**2.5 验证Enabled状态:**

self.UI\_Android.verifyEnabled(self.Pages.Page\_appSettings.checkBox\_usageTracking().isEnabled())

**2.6 验证Selected状态:**

self.UI\_Android.verifyUnselected(self.Pages.Page\_appSettings.checkBox\_usageTracking().isSelected())

**2.7验证数量**

self.UI\_Android.verifyCount(self.Pages.Page\_appSettings.checkBox\_All\_setIndex().getItemsCount(), 3)

self.UI\_Android.verifyCount(self.Pages.Page\_appSettings.checkBox\_All\_noIndex().getItems().getItemsCount(), 3)

**2.8 匹配**

针对<xpath>不熟悉的人，可以用<text>这个标签来简化locator的书写。

<element name="text\_HPSuppliesShopping1"><xpath>//android.widget.TextView[@text = 'HP Supplies Shopping']</xpath></element>  
<element name="text\_HPSuppliesShopping2"><xpath>//android.widget.TextView[contains(@text, 'HP Supplies Shopping')]</xpath></element>

对于以上两个xpath，下面的3种text标签的写法会得到等价的结果

<element name="text\_HPSuppliesShopping3"><text>HP Supplies Shopping</text></element>  
<element name="text\_HPSuppliesShopping4"><text>HP Supplies S.\*</text></element>  
<element name="text\_HPSuppliesShopping5"><text>++HP --ABc++Shopping</text></element>

第二条中**HP Supplies S**.\*代表找到匹配**HP Supplies S**的元素.

第三条中++**HP** --**ABc**++**Shopping**代表找到找到包含**HP** 同时**不**包含**ABc** 同时包含**Shopping**的元素。

**2.9 滑屏**

self.UI\_Ios.swipeUpFromBottomToTop(5, 10)

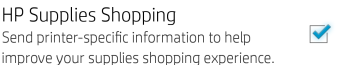
方法是屏幕底部往上滑, 第一个参数是代表从屏幕底部滑到离屏幕顶部还有5 %屏幕纵向距离的地方，第二个参数是代表从离最底部还有10%纵向距离的地方往上滑。

**2.10 拖拽**

self.Pages.Page\_personalizeIcon.buttonOrder\_scanToEmail().dragToElement(self.Pages.Page\_personalizeIcon.buttonOrder\_scan())

把元素scanToEmail拖拽到元素buttonOrder的位置

**2.11 通过附近唯一的元素来确认一个属性不唯一的元素**

****

例如在某些页面checkbox的属性不是唯一的，**难以**直接定位, 而且对复杂的xpath不熟悉时可以用下面的方面来唯一定位.

self.Pages.Page\_appSettings.checkBox\_usageTracking\_forNearby1().getByNearbyUniqueElement(self.Pages.Page\_appSettings.text\_HPSuppliesShopping\_forNearby1()).waitForShown().click()

通过checkBox\_usageTracking\_forNearby1附近的text\_HPSuppliesShopping\_forNearby1元素来唯一确定checkBox\_usageTracking\_forNearby1

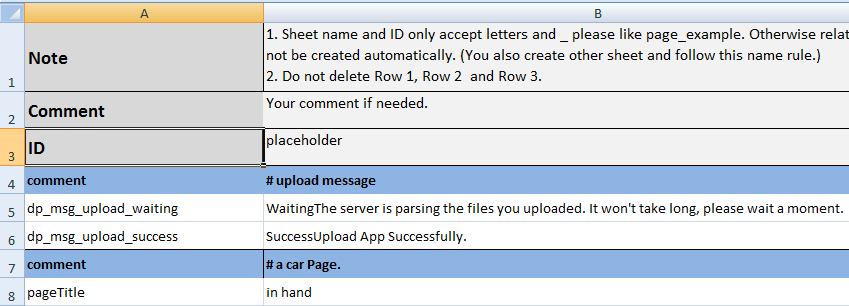
uimap定位写法:

<element name="checkBox\_HPSuppliesShopping\_forNearby1">  
 <id>android:id/checkbox</id>  
 <class\_name index="1">android.widget.CheckBox</class\_name>  
</element>  
<element name="text\_HPSuppliesShopping\_forNearby1">  
 <text>HP Supplies Shopping</text>  
 <text index="1">HP Supplies Shopping</text>  
</element>

可以根据实际情况添加索引属性。

**3 获取testData:**

**如果需要，可以将测试数据保存在testData.xlsx文件中 ：**

****

**在根据你的测试对象调用一个相应的以下方法后**

cls.UI\_Android.loadTestDataFromExcel()

cls.UI\_Ios.loadTestDataFromExcel()

cls.UI\_Web.loadTestDataFromExcel()

**有2个方式可以获取测试数据.**

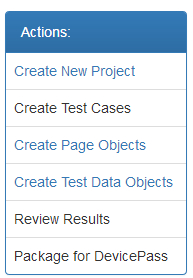
**2.3.1**

**直接从Excel表中拿取ID，然后调用下例方法:**

testData\_string = cls.UI\_Android.getTestData("dp\_msg\_upload\_waiting")

**2.3.2**

**点击 Create Test Data Objects**

****

**生成相应的TestData\_Android, TestData\_Ios和 TestData\_Web类， 然后所有的测试数据可以直接点出来, 例如:**

cls.TestData\_Android = TestData\_Android(cls.UI\_Android)  
testData\_string = cls.TestData\_Android.Sheet\_example.dp\_msg\_upload\_waiting()

**3 手机Web测试:**

**修改相应的runTime.conf文件 ：**

**Android将**

app.path=./../../../app/apk/PrinterControl\_debug.apk  
app.package=com.hp.printercontrol  
app.activity=com.hp.printercontrol.base.PrinterControlActivity  
app.waitActivity=com.hp.printercontrol.base.PrinterControlActivity  
#mobile.browser.name=chrome

**改为**

#app.path=./../../../app/apk/PrinterControl\_debug.apk  
#app.package=com.hp.printercontrol  
#app.activity=com.hp.printercontrol.base.PrinterControlActivity  
#app.waitActivity=com.hp.printercontrol.base.PrinterControlActivity  
mobile.browser.name=chrome

**Ios 将**

app.path=./../../../app/app/RemoteControl.ipa  
app.bundleId=com.hp.printercontrol  
#mobile.browser.name=safari

**改为**

#app.path=./../../../app/app/RemoteControl.ipa  
#app.bundleId=com.hp.printercontrol  
mobile.browser.name=safari