|  |
| --- |
|  |
| 自动化测试(Android) |

自动化测试(Android)

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 事件 | 时间 | 责任人 |
| 1.0 | 创建 | 2016-04-05 | 廖正阳 |
|  |  |  |  |

基于windows7 64位操作系统

# 环境搭建

## 工具安装和环境变量配置

#### 安装node.js

#### 安装Appium

按提示安装缺失的工具

示例路径：C:\Program Files (x86)\Appium\node\_modules\.bin

#### 安装java环境，以jdk1.7为例

示例路径为：A:\Java\jdk1.7.0\_13和A:\Java\jre7

#### 安装Android SDK

示例路径为：A:\DevTools\android-sdk-windows

#### 安装Android模拟器

Android SDK自带模拟器太卡

#### 安装python, 直接复制文件包到指定目录。

示例路径为：C:\Python27

#### 安装python IDE，推荐Pycharm

#### 相对应的环境变量配置为：

**JAVA\_HOME**

A:\Java\jdk1.7.0\_13;

**PATH**

A:\Java\jdk1.7.0\_13\bin;A:\Java\jdk1.7.0\_13\jre\bin;%ANDROID\_HOME%\platform-tools;%ANDROID\_HOME%\tools; C:\Python27;C:\Python27\Scripts; C:\Program Files (x86)\Appium\node\_modules\.bin

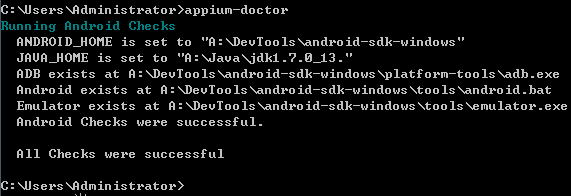
**CLASS\_PATH**

A:\Java\jdk1.7.0\_13\lib\dt.jar;A:\Java\jdk1.7.0\_13\lib\tools.jar;

**ANDROID\_HOME**

D:\DevTools\android-sdk-windows;

打开命令行窗口，输入appium-doctor，检查环境配置是否成功



图：1-a-1

如图1-a-1所示，为环境配置成功。

## 工程配置

#### 使用IDE导入ATFramework

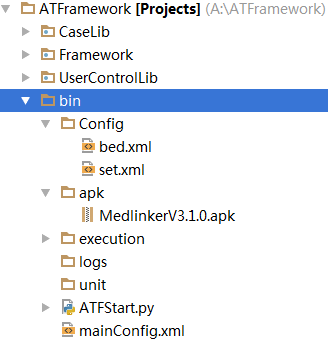


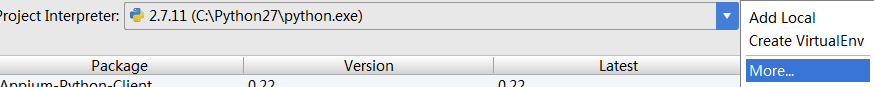
图1-b-1

另外便于后期命令行启动appium服务端和自动化测试，请在PATH环境变量中添加

“; C:\Program Files (x86)\Appium\node\_modules\.bin; ”，修改python安装路径下：C:\Python27\Lib\site-packages\MyPath.pth为ATFramework实际路径 。

#### 配置解析器

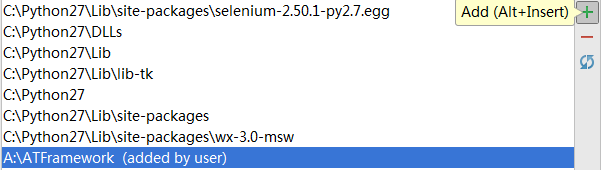
IDE >> File >> Settings >> Project >> Project interpreter >> 选择python安装路径下的python.exe；



选择more；



选中配置的解析器，点击图中红框部分；



将ATFramework的路径添加到解析器包搜索路径；

>>应用。

# ATFramework

ATFramework目录下工4个目录或package。CaseLib主要放置测试集合测试用例和每个测试集自己使用的接口；Framework主要定义了驱动测试集合测试用例class和接口、日志打印接口和可能公用到的方法；UserControlLib封装了产品的主要功能接口；bin放置测试执行的入口文件，主配置文件，测试需要的app安装文件，测试床和测试集配置文件和默认的自动化执行日志保存路径。

## 主配置文件

事例为bin目录下mainConfig.xml；



图: 2-a-1

定义了测试执行参数，测试执行所依赖的测试床和测试集配置；格式固定，详细说明如 图：2-a-1。

## 测试床

事例为bin\Config\bed.xml；



图: 2-b-1

定义了测试执行需要的所有设备信息；格式暂时固定，详细说明如 图：2-b-1。

## 测试集

事例为bin\Config\bed.xml；



图：2-c-1

指定测试执行的测试套和测试用例；格式暂时固定，详细说明如 图：2-c-1。

## 启动自动化测试

#### 启动参数

参数说明：

**--configFile: (required) The main test configuration file.**

**--testBedFile: The Testbed xml file.**

**--testSetFile: The testSetFile xml file.**

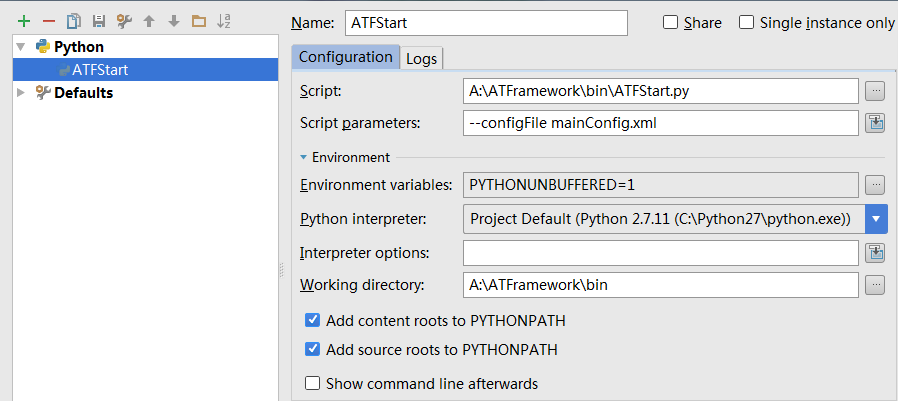
**--localExecutionLogPath: The local execution log path.**

configFile为必选参数，其余参数可选。

#### IDE启动方式

配置参数

IDE >> Run >> Edit configurations >>



>> OK >> Start or Debug

#### 命令行启动

命令行切换到ATStart.py所在路径，执行”ATStart.py --configFile xxxx.xml”

# UserControlLib

UserControlLib是针对具体被测对象和业务抽象和封装。

## 对Android设备的抽象-AndroidPhone

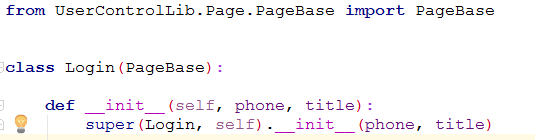
生成Android对象时，使用appium链接一个真实或虚拟的Android设备，并安装了被测app；进入到Welcome页面。

## 对Page的抽象-PageBase

每一个page对于单个AndroidPhone是单例的。每个页面对应一个xml配置文件。每个xml配置文件定义了当前页面的可操作元素以及每一个事件生成的新页面对象。doAction实现对元素的操作并等待定义的响应完成。

#### 定义当前页面class

例如定义个Login类，module(UserControlLib.Page.Page.Login.Login):

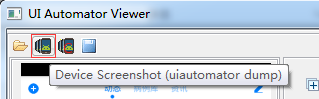


对应xml为UserControlLib.Page.Element.Login下的Login\_password.xml和Login\_code.xml

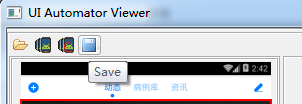
#### 定义Page对应xml配置

获取当前页面的元素数据

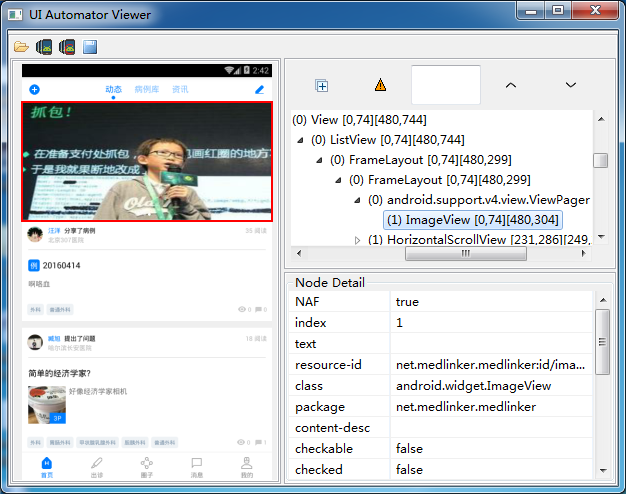
**>>**打开android-sdk-windows\tools\uiautomatorviewer.bat，获取当前页面数据，



**>>**导出当前页面数据，数据文件为uix文件；



**>>**选中要操作的当前页面中的元素，根据选中元素的属性值，定位uix文件中元素的数据，

>>在元素数据中定义alias属性（测试脚本中使用），并定义此元素操作对应的响应，

**alias="account\_frame" swipe\_left="Login\_password"**

或

**alias="start" click="Home\_dynamic|Login\_code"**

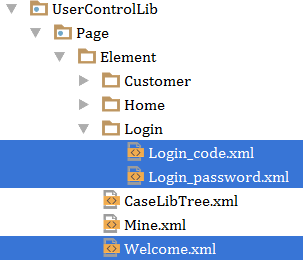
alias指定别名，事件(click，swipe\_left，swipe\_up，swipe\_down，swipe\_right，double\_click)可能使app切换到的页面，根据primary判断实际到达的页面。

**>>**定义当前页面的primary元素，以判断实际是否为当前页面，(建议定义在有alias的元素上)，

**alias="username\_edit\_text" primary="true"**

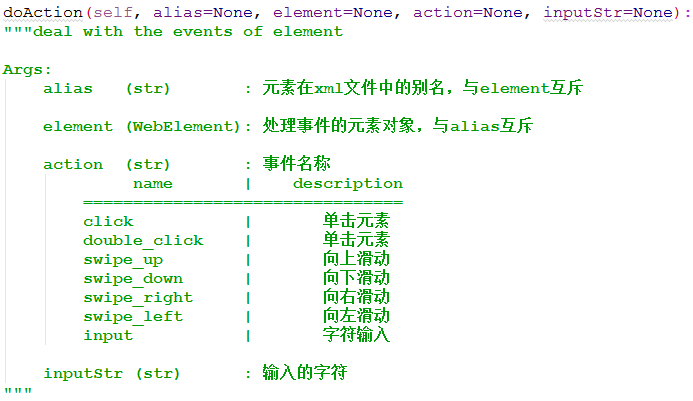
属性值为true，一个页面可以有多个primary。

另存为‘页面名称.xml’如Welcome.xml或‘页面名称\_子页面.xml’如Login\_code.xml；放置到UserControlLib.Page.Element下。

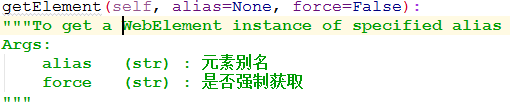


## doAction和getElement

doAction操作当前页面指定的WebElement或指定别名的WebElement，并等待指定的响应完成，错误则报错。



getElement获取当前页面指定别名的WebElement，可强制获取。



# 测试用例的实现

用例

ID：Mine\_info\_integrity

Description: 完善资料完整度

preHandle:

1.登录App，进入我的-首页

testHandle:

1.点击资料完整度，进入完善资料页面;

2.点击返回，返回到我的-首页;

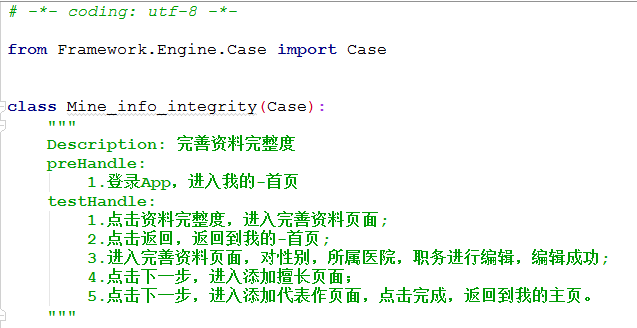
3.进入完善资料页面，对性别，所属医院，职务进行编辑，编辑成功;

4.点击下一步，进入添加擅长页面；

5.点击下一步，进入添加代表作页面，点击完成，返回到我的主页。

>>新建脚本Mine\_info\_integrity.py；

>>实现Case；



## import Case基类；

from Framework.Engine.Case import Case

## 定义类Mine\_info\_integrity，继承Case；

## 完善类文档;

即用例说明，便于日志说明当前case的操作和测试点；

## 实现方法主要测试函数

preHandle，testHandle和postHandle；preHandle和testHandle的操作在用例描述里有详细说明，postHandle要根据用例描述，清除当前用例对后续用例的影响。建议当前页面到指定的主页面，即调用方法arriveSpecMainPage。

例: 1.点击资料完整度，进入完善资料页面;

根据用例的preHandle说明，preHandle的实现：

**def** preHandle(self):  
 self.logStep(**'1.登录app，默认展示首页;'**)#测试步骤的日志说明  
 self.phone = self.resource.getAndroidPhone(id=**'0'**) #获取android对象  
 self.phone.login()#按测试床配置的账号登录app  
 self.phone.arriveSpecMainPage(main=**'mine'**)#切换到“我的-首页”

当前页面处于‘我的-首页’；

脚本中需要对元素‘资料完整度’点击。

>>按照3.b定义page的xml配置文件；

>>对元素别名为alias=“info\_integrety”；事件click=“InfoIntegrety\_1th”；

>>添加配置文件InfoIntegrety\_1th.xml，定义后续使用到的元素alias和primary属性；

>>

self.logStep(**'点击资料完整度，进入完善资料页面'**)  
self.phone.doAction(alias=**'info\_integrety'**, action=**'click'**)

# 常见问题及处理方法

#### 元素text属性为中文，与预期值进行比较；

self.phone.getSpecElementText(alias=**'sex\_text'**) != **u'男'**

元素text属性值为中文时，获取后为unicode类型，进行比较是对预期值进行Unicode转换

#### 某元素属性text，不是固定值

获取和操作元素时，xpath不能指定text属性，使用text=‘ignore’过滤掉text属性