

AUTOMADB

**Core Documentation Version 2.0**

Joaquin Miranda Castro

192493

The following document explains the different folders, libraries and methods which are used within the automation framework.

Folders

* docs: This directory contains all of the documentation for the automation framework. Including the business case (architecture within document, release notes, setup to execute scripts, environment etc.
* qa: This folder contains documents related to QA which include test plan (test strategy), test cases, traceability matrix, reports and execution logs.
* src: This directory contains the scripts which are divided into different folders:
  + lib: contains different libraries which holds the methods utilized in the scripts.
  + scripts: contains the scripts which contain the end to end flows for the execution of the test cases, to test the functionalities for the applications under test of the mobile device. The methods in the libraries are used in this folder.
  + suites: contains the definition of the test suites, which are essentially various test cases which have a common functionality, which are called in conjunction to have a more robust test.
  + devices: contains json definitions of different android version, which allows the framework to be compatible to different devices. To allow a new device, a new file must be added with the uiautomator names of different objects that are used in the scripts. Then, the name of the file must be associated to a serial number in the device\_compatibility.json and after that the execution will be able to recognize the device and associate it with an android version.
* inputs: contains the inputs for some of the test suites. For v2.0, holds the inputs for the test cases in the Calculator App Test Suite.

Methods:

The following are methods defined in the different libraries.

* Logger: this library is in charge of showing the data in console and in the file:
  + begin\_log: Starts a new log in de specified file.
  + write\_log(mensaje): Writes log and in console displayed received message.
  + end\_log(): At the end writes a report with the initial and final times.
  + error\_log(mensaje): Writes the error message.
* Utils: this library handles simple methods that can be reused, but do not have such a specific task that requires a specific library to be created for each.
  + validate\_number(number): Validates the phone number being dialed. This validation recognizes the emergency number 911, national Bolivian phone numbers and international numbers with the + prefix.
  + read\_json: reads any son file and converts it to a dictionary
  + get\_device\_data: determines the version of the android based on the serial number after reviewing a specific file.
  + is\_close: compares if two floats are almost identical, due to the fact that python and android sometimes return different results in decimal numbers.
* PhoneControl: This library provides the methods to connect the phone via ADB and UI Automator. It also has the tool necessary to execute a variety of actions within the mobile device.
  + read\_serial: Identifies the dies serial number within the adb list.
  + unlock\_phone: Unlocks the cellphone as long as there is no security measure such as a password.
  + click\_home: Simulates pressing the home button.
  + click\_back: Simulates pressing the back button.
  + call\_adb: makes a call in the phone using adb only.
  + check\_call\_adb: checks the phone state to determine if the phone is in a call, using the adb shell. Returns true if the state is 1 (dialing) or 2 (answered call), and False elsewhere.
  + end\_call\_adb: ends a call using adb shell.
  + switch\_button(texto): Executes a click to a switch which contains the text passed as a paramater and which className is android.widget.Switch.
  + click\_button(texto, className): Executes a click to a button that matches the text and className parameter.
  + longclick\_button(texto, calssname): Executes a long click on a button that matches the text and className parameter.
  + button\_exists(texto, className): Identifies if a button that matches the text and className currently exists.
  + info\_select(packageName, className): obtains the text parameter of an object that matches the packageName and className.
  + info\_select\_detailed(classname, packagename, resourceid): obtains the text parameter of an object that matches the resourceid, packageName and className.
  + button\_checked(packagename, classname, resourceid): ): returns true if a button or switch is checked, while it matches the resourceid, packageName and className.
  + detailed\_button\_exists(classname, packagename, description): checks if an object that matches the parameters exists in the current ViewPort.
  + click\_detailed\_button(className, packageName, description): Executes a click to a button that contains the three parameters which are passed.
  + longclick\_detailed\_button(className, packageName, description): Executes a long click to a button that contains the three parameters which are passed.
  + set\_text\_textfield(classname, packageName, contenido): modifies the textfield that matches the parameters with the content passed in as well.
  + get\_text\_textfield(classname, packagename): retrieves the content of the textfield that matches the parameters given.
  + clear\_text\_textfield(classname, packagename): cleans the textfield that matches the parameters given.
  + version: using an adb shell command, returns the android version of the device currently being used in the framework.
* Calculator: This library provides the methods regarding the calculator application, validating the results and calculating the desired operations.
  + calculate: executes any of the 4 operations, the digits of precision can be defined, only 2 numbers can be used for a calculation.
  + validate\_digits: for every number it validates that it cannot contain more than 15 digits, and no more than 10 decimal numbers.
  + validate\_number: validates that the input are numbers.
  + division: executes the division operation with two numbers.
  + addition: executes the sum operation for two numbers.
  + subtraction: executes the subtraction operation of two numbers.
  + multiplication: executes the multiplication operation for two numbers.
  + validate\_result: comprares the result of ui automator with the expected result that was executed in python with a certain margin of error to accommodate for the different android based calculations.
* STF: This Library provides methods to connect to and SFT Instance running, and to be able to connect to the devices in that Android Device Farm through their API.
  + get\_all\_stf\_devices: gets the serials of all devices connected to STF.
  + stf\_connect(serial): activates the STF device that matches the serial and gets the remote url for connecting to it.
  + stf\_individual\_connection(serial): changes the state of the device to using=true through the API.
  + get\_remote\_serial(serial): obtains the remote url for an specific serial. This allows the local adb to connect to it.
  + connect\_remote(url): connects the local adb to the remote device.
  + disconnect\_remote(url): disconnects the local adb server from the remote device.
  + stf\_disconnect(device\_serial, device\_remote\_serial): disconnects the STF device and disables remote debugging for it.
  + stf\_individual\_disconnection(device\_serial): stops using the remote device, and changes its parameter ‘using’ to false.
  + remove\_remote\_serial(device\_serial, device\_remote\_serial): tells STF to stop considering this device to be in remote debugging.