**Description:**

This project is a POC that is created to demonstrate Mobile Automation performed with the help of “**RobotFramework**” and “**Python**” technology. This POC contains a basic user workflow of the user ***Log-in*** and ***Buy Equity***.

For demonstration purpose this POC uses **Upstox pro** public *Android app*. This test can seamlessly be performed on any other iOS/Android apps with minor script changes in the workflow and configuration.

**Tools and Technology Used:**

1. Python 3.7 and above (Robot Framework works with this version of Python)
2. Appium
3. Android Studio
4. Robot Framewok
5. Node.js
6. Robot-Appium Library
7. PyCharm (Python Editor)

**Prerequisite:**

* **Environmental:** **Environmental Variable** is already configured for
  1. Java home
  2. Android Home
  3. Node
  4. Appium Library
  5. PyCharm
  6. Robot Framework
  7. Python Home
  8. Android Apk (On which the test has to be performed)
* **Application: Test data** should be available for the desired workflow
  1. User credentials are already available
  2. Test Data is already present
* **Robot Automation Script:** Robot script needs some **device** and **apk** **configuration** to

run the scripts in perfect way.

* 1. File: *Utility\_setup.robot*
     1. ${automationName} : Appium
     2. ${appium\_host} : http://localhost:4723/wd/hub
     3. ${platformName} : Android
     4. ${platform\_version} : 9 (Depends on Device)
     5. ${deviceName} : **emulator-5554** (Based on Real or Emulator)
     6. ${appPackage} : in.upstox.pro
     7. ${appActivity} : .views.main.UpstoxLauncherActivity
     8. ${app} : “$local/apk/path”
     9. ${alias} : upstox

**ROBOTSCRIPTS Architecture:**

Robot Framework is basically aligned for acceptance **Test driven development** (ATDD), along with that this POC is designed based on the **Page object modeling** as each screen is dived into module. Along with that this POC behaves in **Keyword driven modeling**. So altogether it is the complete Hybrid solution for Mobile Automation.

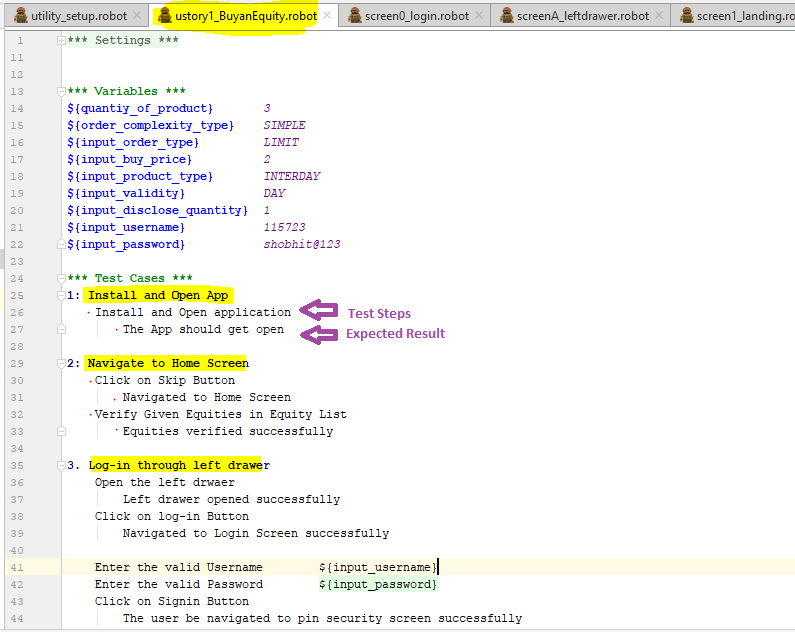
The project for the POC structure is as follow:

**MyPOC >> testsuite**

* ustory1\_BuyanEquity (Test Script): This is the user story based test script, where you find one user story is implemented with the help of screen resources file.

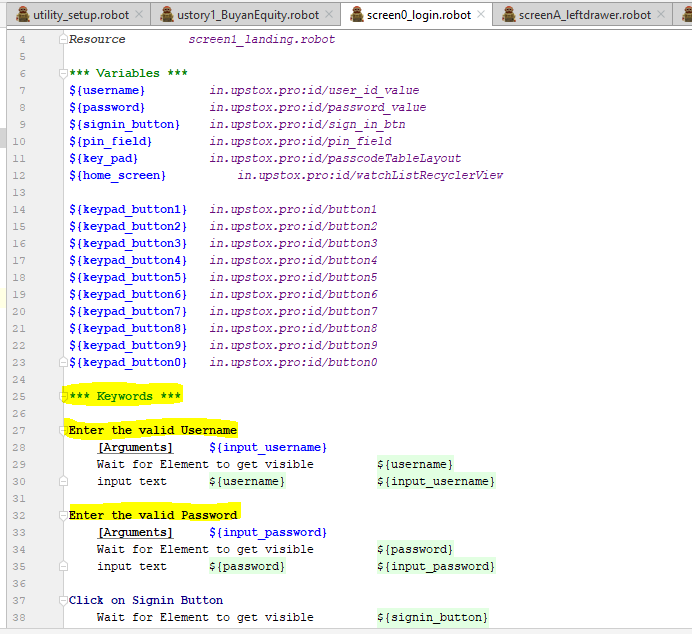
The test steps are written in such a way that anyonecan get the idea what is being executed.

e.g.:

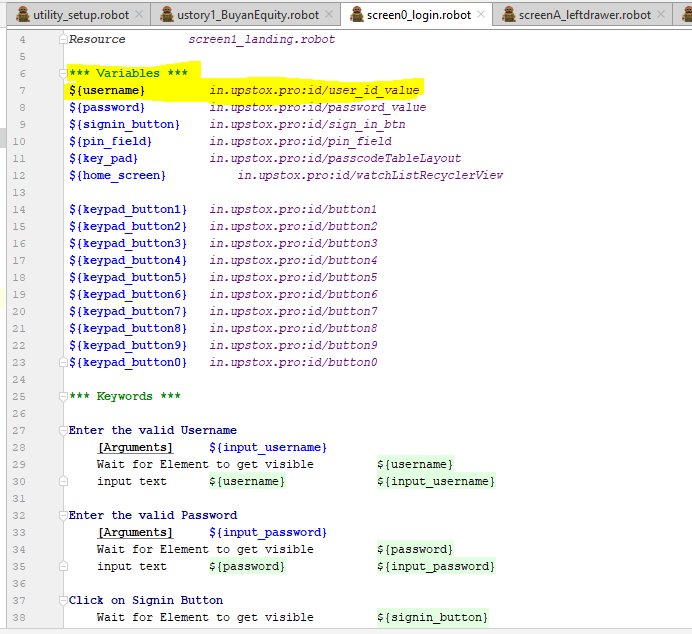


* resources : This contains all the resource file for each of the screen, the resources consists of

**Keyword**: This will be the user customized keywords for the test cases that are created in a way that anyone can get the idea what the functionality is meant to do.

e.g.: 

**Variables**: These are the elements and data defined, which are used in the *Keywords*

e.g.: 

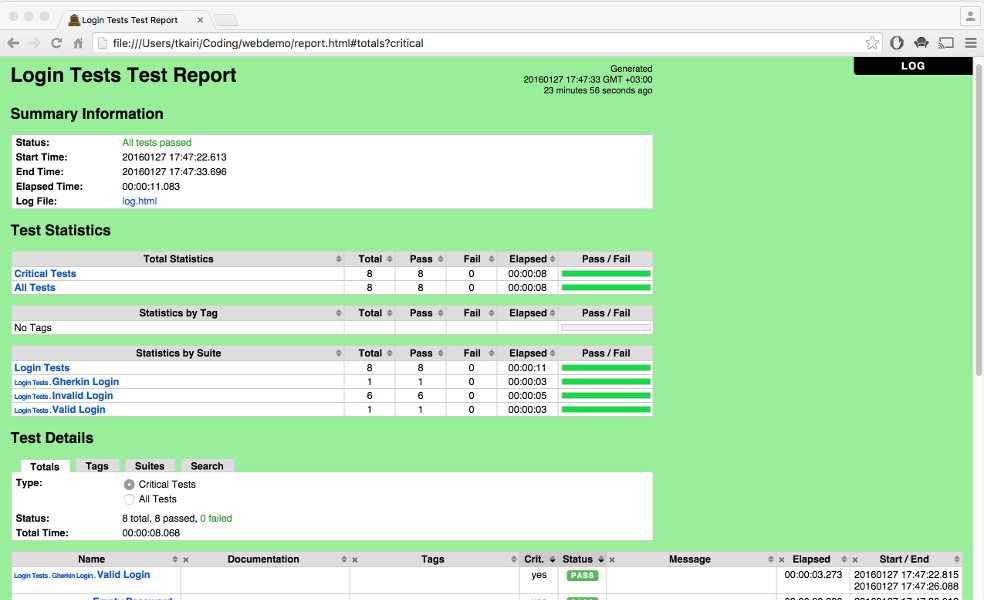
**Resource files for the POCs are:**

1. *Base Environmental variables and set-up file*
   1. utility\_setup.robot
2. *POM based Resources:*
3. screen0\_login.robot
4. screen1\_landing.robot
5. screen2\_home.robot
6. screen3\_equityports.robot
7. screen4\_buyEquity.robot
8. screen5\_reviewAndbuy.robot
9. screenA\_leftdrawer.robot

**Results:** All the results generated by the script will be stored in the directory in different formats with failure **Screen Screenshot** if any.

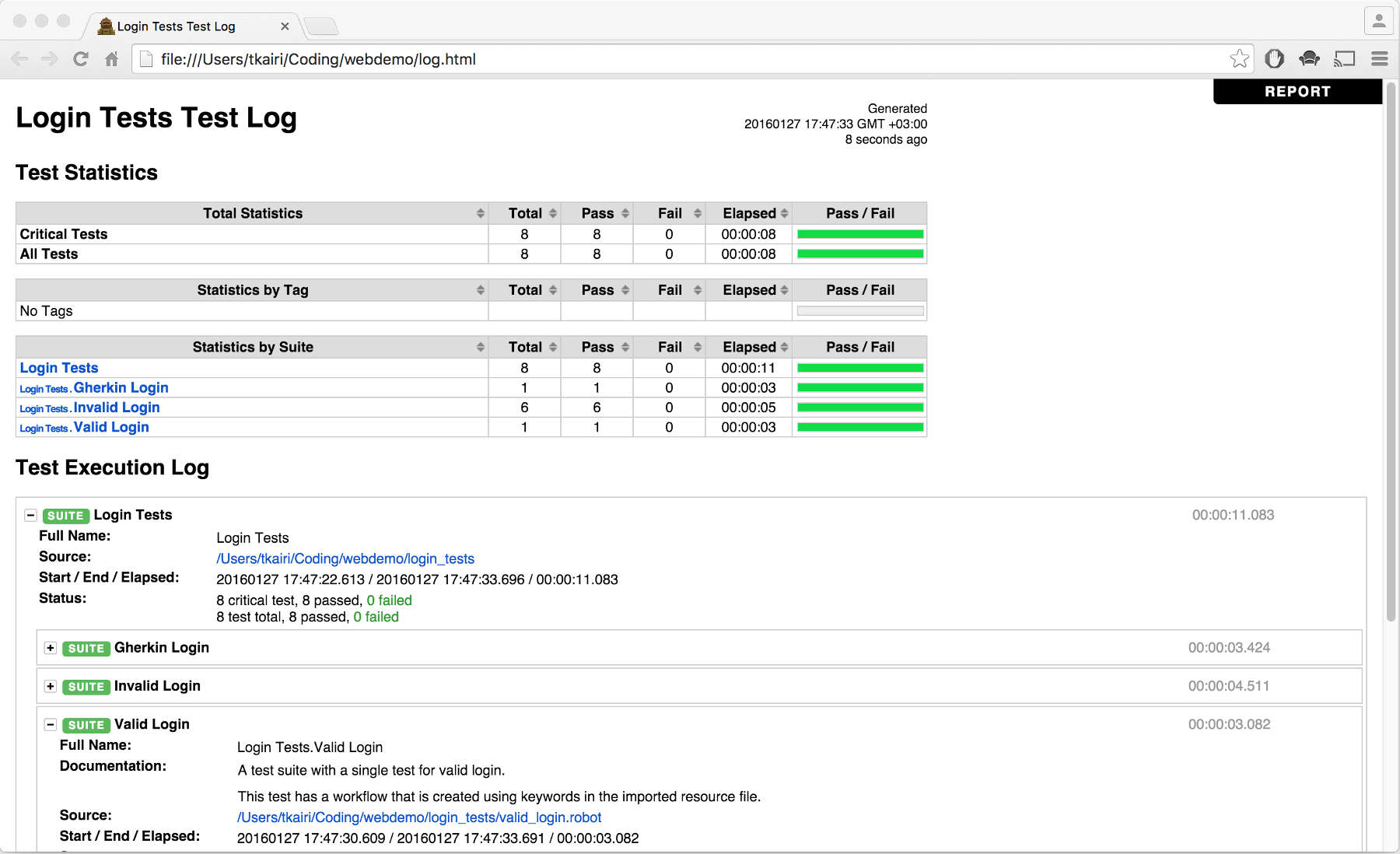
Reports.html

e.g:



Log.html

e.g:



**Script Execution:**

Below is the process that gives you the idea how to execute the scripts.

**Prerequisite:**  Considering some basic aspects that are already configured and present, those are:

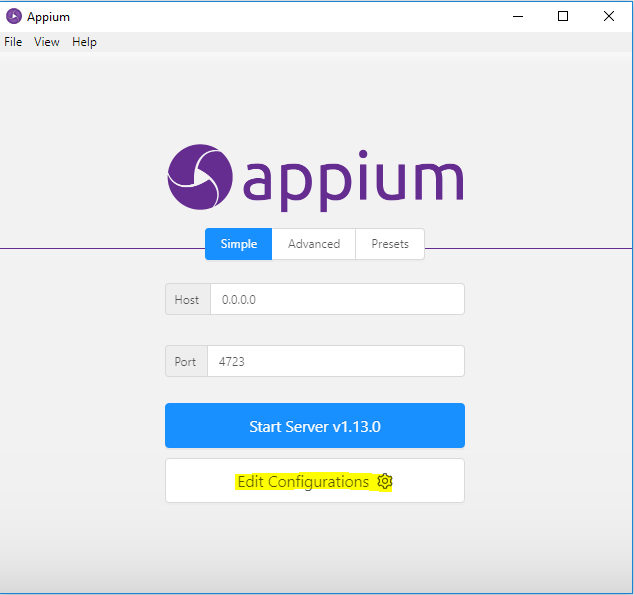
1. All the tools and its path setup
2. Test data is already present
3. Update the pip using:

* pip install --upgrade pip

1. Install Robot Appium-Library using:
   * pip install robotframework-appiumlibrary
2. Start the emulator

* Create a device with latest configuration and mention the same in the **utilites\_setup.robot** file.
* In case of Real device mention the device name and configuration.
* To get the device name use **adb devices** command on cmd prompt/terminal.

1. Configure the Appium for local host and Android Home



1. Start the Appium at local host.
2. Run the Robot script using command:

Robot -T –d <result-directory-name> <script-path>.robot.

* -T : Generated report with Data and Time stamp
* -d : For creating results directory

**Note: The POC is created in a short duration hence work is in-progress mainly for**

* Code optimization
* Script performance
* Script Designing
* Test cases for iOS app will be added soon

Rest of the things will also be taken care when it is completed fully.