**AUTOMATION FRAMEWORK README**

# Introduction:

This framework is used to automate test case for Web UI. It was developed by using Robot Framework and Python. It allow user:

Executing tests in multiple browser (Chrome and Firefox)

Executing test parallel by using command line

# Design structure:

TestCases

keywords

Pages

Executor

Lib

* **Pages directory:** where contains locator for elements. Each class for each page
* **Lib:** where contains my custom library such as browser factory, data reader.
* **Keywords**: where contains keywords for testing pages. Each class for each page
* **Test** **case**: Where contains test suites and test cases. The test case will call created keywords in keywords directory

# Implement Guidelines

## Pre-required package:

* Robot Framework 4.0
* Python 3.9.2
* robotframework-selenium2library 3.0.0
* robotframework-seleniumlibrary 5.1.1
* pandas 1.2.3
* openpyxl 3.0.7
* xlrd 2.0.1
* webdrivermanager 0.10.0

## Steps to implement

Note: the following steps is for Windows 10. For other OS, the steps may be different a little bit but general steps should be same

**Step # 1** – Install Python and PIP

* Go to python.org (https://www.python.org/downloads/) and download desired version
* Execute downloaded installer
* Check ‘Add Python to path‘



* Click Install Now

**Step # 2** – Install Robot framework and Selenium library through PIP

* Use this command ‘pip install robotframework‘ for robot framework library
* Use this command ‘pip install robotframework-seleniumlibrary‘ for selenium library

**Step # 3** – Install required library through PIP

* pip install pandas
* pip install openpyxl
* pip install xlrd

**Step # 4** – Install webdrivermanager and download latest webdriver for Firefox and Chrome

* pip install webdrivermanager
* webdrivermanager firefox chrome --linkpath /path/to/driver
* Add path to driver into Environment Variable PATH

**Step # 5** – Get code from github <https://github.com/ntnguyenbao/ntnguyenbao_vdc_tech_test>

# Test execution guidelines

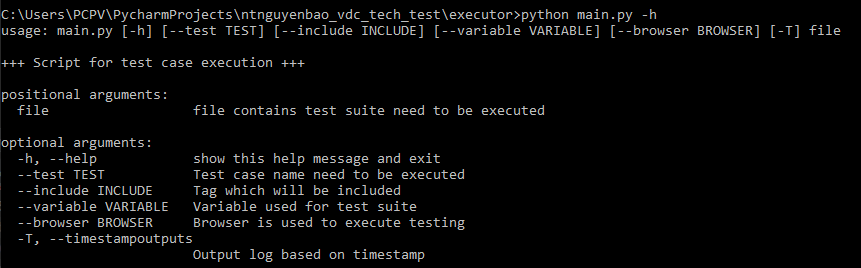
## Test execution

First of all, we need to change working directory to ntnguyenbao\_vdc\_tech\_test /executor

* cd ntnguyenbao\_vdc\_tech\_test /executor

For list of execution parameters

* python main.py --help



For executing all test cases in file

* python main.py ..\TestCases\test.robot

For executing specific test case

* python main.py ..\TestCases\test.robot --test <test case name>

*(Ex: python main.py ..\TestCases\test.robot --test Func-001)*

For executing tests in specific browser

* python main.py ..\TestCases\test.robot –browser <firefox/chrome> (by default, Chrome will be used)

*(Ex: python main.py ..\TestCases\test.robot –browser firefox)*

For executing tests which belong to specific Tag:

* python main.py ..\TestCases\test.robot --include <tag name>

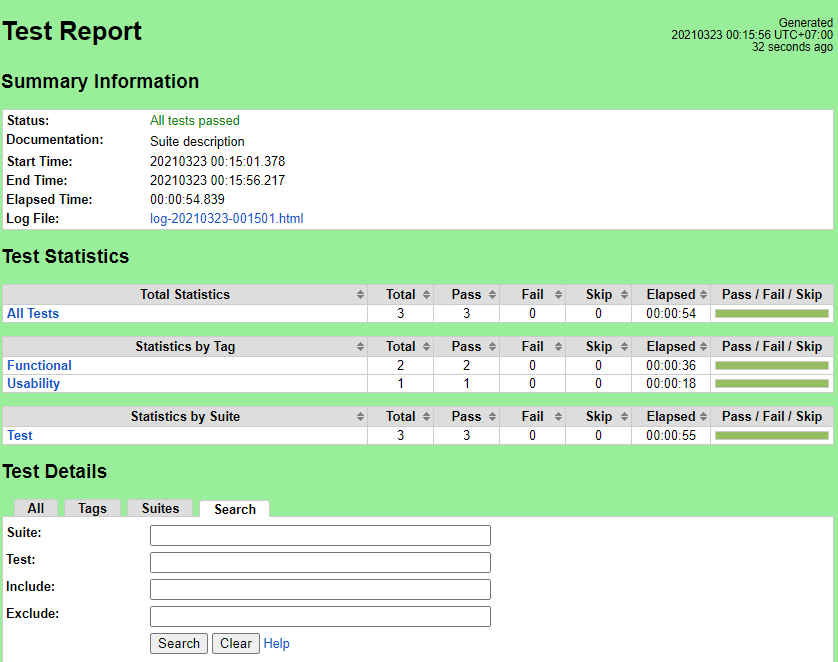
*(Ex: python main.py ..\TestCases\test.robot --include Usability)*

## Test Report

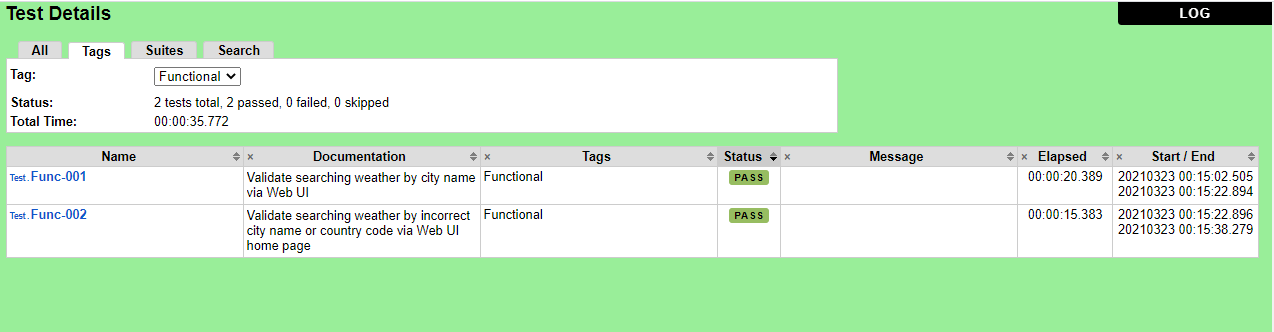
The report will be generated by built-in module of Robot Framework. The report will be saved in HTML format with links to pages

The report output files will be located in ntnguyenbao\_vdc\_tech\_test /report/ddMMYYYY

**Report main page**



**Test Detail**



**Test Log**

