**Raw Innovations Assessment**

**Author:** Nandhakumar Rajendran

**Date :** 06-04-2022

**Description:** This is a code challenge for prospective QA Automation Engineer to work for Raw Innovation Sweden/VideoSlots.

**Task Description:**

Imagine you are a QA Automation Engineer working with the renowned online notes taking Evernote. You are assigned to build an automation framework from scratch, so to build up all the required E2E automation tests, including UI and API testing.

Tests to be covered:

1. Unsuccessful login using email

2. Successful login using email

3. Login and write a note followed by a logout

4. Login again and make sure you open the note create in step 3

**Solution Implementation:**

1. The Test Automation script development is handled in Robot Framework Automation based on Python Language

2. The Script is handled based on page object Model

3. This Script Approach includes both Behaviour Driven Development and Keyword Driven Development

4. Individual Step files for both login and Note Creation Test Cases

5. Execution is done via Argument File Functionality in Robot Framework

Details on Test Folders & Files as follows:

* **CommonLibrary** – This Folder has all the common functional libraries could be used across different tests
  + **StringManipulation.robot** - Generic Library for All reusable String Manipulation Keywords
* **Pages** – This Folder has all the individual pages with its corresponding Page Objects and Keywords
  + **dashboard\_page.robot** - This File Contains all the Page Objects and Keywords for Evernote Dashboard Page
  + **home\_page.robot** - This File contains all the Page Object Elements and Keywords for Evernote Home Page
  + **login\_page.robot** - This File Contains all the Page Objects and Keywords for Evernote Login Page
  + **signout\_page.robot** - This File Contains all the Page Objects and Keywords for Evernote Signout Page
* **Steps** – This Folder has all the test steps file with test steps required for test cases
  + **Login\_steps.robot** - This File contains all the steps required for the login test
  + **Notes\_steps.robot** - This File contains all the steps required for the Note Creation test
* **tests** – This Folder has all the test suites for test cases with Login and Notes Creation
  + **Login\_tests.robot** - Test Suite file to validate the Login test cases
  + **Notes\_tests.robot** - Test Suite file to valid the Dashboard-Notes cases
* **result** – This Folder has the test result reports and log files for test executions
* **resources.robot -** This File contains all the necessary resources and application Details
* **ExecutionConfigFile.txt** – This is a argument file for the test scenarios executions
  + **Input Variables –** Parametrizable
  + **Test Suites** to be Executed in sequential order

**How to Execute:**

1. Install Python from Python.org with path added to system environment variable
2. Download and keep the Chromedriver in local machine location. Add the path to environment Variable
3. Install the below robot framework libraries
   1. robotframework
   2. robotframework-seleniumlibrary

How to Install:

* Open Cmd in admin mode
* Use command – “pip install <<libraryname>>”
* Eg., pip install robotframework

1. Save the test scripts folder “Raw Innovations to Local”
2. Open cmd and Navigate to Raw Innovations Folder
3. Type “robot –outputdir /results –A ./ExecutionConfigFile.txt” and press Enter

**Test Result Validation:**

1. The test results with pass or fail status will be updated for every test case after execution in the command line
2. Final test results can be seen in the Log.html, report.html and Output.xml file
3. Overall test results with all the statistics of the whole executions can be seen in report.html
4. Individual step level test results can be seen in log file

**Solution Proposals:**

1. Data Driven Approach - Input and Output can be done via external file (eg. Excel) operations – if its required for larger implementations
2. Execution can be done by one robust Driver script written in Robot to execute multiple test suites in sequential order(Gui testing, API testing via postman/newman)
3. Cross Browser testing can be done just by parameterizing the Browser name in Resources.robot
4. I can also develop the same test scenarios automation test in core Python via Pytest using selenium
5. I have done web automation with java – Selenium in past. Can also do in future if it required in java

**Key points and Interests:**

1. **I have solved this assessment in python, but I can also learn and automate in java, javascript also if its required for automation projects in Raw innovations projects**
2. **I also have interest in learning other automation tools in market**
3. **I have experience in implementing Azure for Automation respository and scheduled nightly executions**
4. **Expertise in defining different automation approach based on the automation requirement**
5. **Being an enabler of automation, I have trained and supported many newbies on Automation both internally in my team and also other product teams in my circle of work.**