AUTOMATED SOFTWARE TESTING
(502072)

Lab 8 – File Upload/Download/Rest APIs
Prepared by Nguyen Thanh Quan (MEng)

1. **GOALS:** This lab helps students to
   - Practise automation testing of web applications using.
   - Be able to write test cases with Selenium Library – Robot Framework.
   - Understand various functionalities of Selenium Library.
   - Be able to work with file download/upload.
   - Be able to work with APIs.

2. **OBJECTIVES**
   - Install automated testing environment with Selenium/Robot Framework.
   - Install web drivers.
   - Write and execute test cases.
   - Work with file download/upload.
   - Work with window operations.
   - Work with APIs.

3. **CONTENT**

3.1 PREREQUISITES

To practise complete this assignment, students must prepare to install the required libraries below which are also presented in previous labs. More importantly, students should have a basic understanding of testing concepts.

To practise complete this assignment, students must prepare to install the following libraries and packages. More importantly, students should have a basic understanding of testing concepts.

- Python
  Go to python official site – https://www.python.org/downloads/ and download the latest version or the prior version of python as per your operating system.
  Remember to set PATH correctly to use Python after installation.

- Pip
  PIP gets installed along with python. Run **`*pip --version*`** to check pip version

- Robot Framework
  Use pip – python package manager to install the robot framework and the command for it is as follows

  `pip install robotframework`
  `robot --version` to check robot framework version.

- wxPython for Ride IDE
  wxPython is needed for Robot Framework Ride, which is an IDE for Robot Framework.

  Windows:

https://sourceforge.net/projects/wxpython/files/wxPython/2.8.12.1/

Linux: Install wxPython with the package manager of OS.

- Selenium library (https://github.com/robotframework/SeleniumLibrary/)
  `pip install --upgrade robotframework-seleniumlibrary`

- Selenium webdriver
  (https://www.selenium.dev/documentation/webdriver/)
  `pip install --upgrade robotframework-seleniumlibrary`

- Robot Framework Ride
  Use pip command to install Ride IDE.

  `pip install robotframework-ride`.

To open Ride IDE, run ***`ride.py`***

3.2 THEORY

Student must review the theory of Robot Framework and Selenium Library which are presented in the previous Labs.

Restful APIs: REST stands for REpresentational State Transfer. REST is web standards based architecture and uses HTTP Protocol. It revolves around resource where every component is a resource and a resource is accessed by a common interface using HTTP standard methods. REST was first introduced by Roy Fielding in 2000.

Restful API references: https://www.restapitutorial.com

3.3 PRACTICE

For testing, it becomes important to understand how to interact with the browser and locate the html elements. It is very easy to work with input fields with robot framework. In this lab, we will learn how to work with form fields namely textbox, radio button, dropdown, checkbox, text area etc using Selenium Library. To work with form field elements, we need the locator, which is the main unique identifier for that the fields and it can be id, name, class, etc.

Let create a new project and import Selenium Library properly.

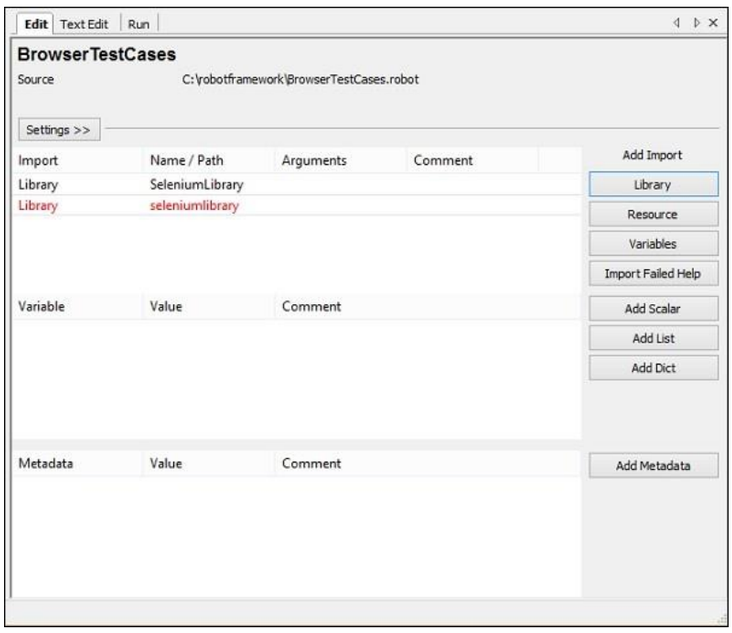

Figure 1 - Valid/invalid library import

The name given has to match with the name of the folder installed in site-packages.

In case the names do not match, the library name will be in red as shown above. Library import in red is as good as the library does not exist inside python. Now, we have completed selenium library import.

The official document page of Robot Framework provides users with detalied explanation and demo, students should refer to the page during the time working with this lab.

https://docs.robotframework.org/docs

https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#introduction

https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html

Locating elements:

https://selenium-python.readthedocs.io/locating-elements.html

Chrome/Edge extensions for locating Xpath:

https://www.browserbear.com/blog/9-best-chrome-extensions-to-find-xpath-for-selenium-and-other-automation-tools/

x/y coordinates:

https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html?_ga=2.47015591.1688393203.1707139684-1841019508.1706956421#Click%20Element%20At%20Coordinates

3.4 REQUEST LIBRARY

In this lab, students must import Request Library to practise API tesing.

Reference: https://docs.robotframework.org/docs/different_libraries/requests

Request Keywods:

https://marketsquare.github.io/robotframework-requests/doc/RequestsLibrary.html

3.5 EXERCISES

Let review and research about locators for form fields element to complete the assignment.

Assignment 1: Open browser

- URL to open: https://practice-automation.com/

Assignment 2: Interact with web elements via mouse with Selenium library

Test case 1:

- Navigate to "File Upload" page by clicking "File Upload" button displayed on the page just opened.
- Click the buttons to choose file to verify file open dialog.
- Set up to upload a file and verify

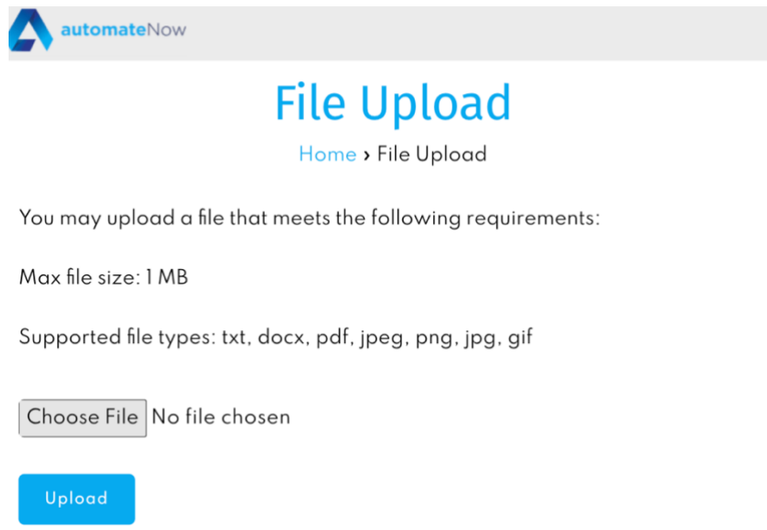

Figure 2 – File upload

Test case 2:

- Navigate to "File Download" page by clicking "File Download" button displayed on the page just opened.
- Click the download buttons and verify the process.

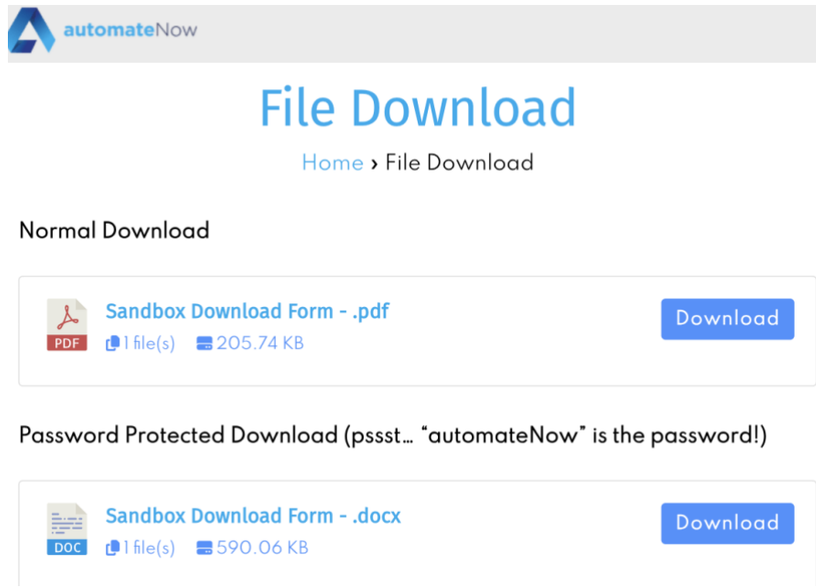

Figure 3 – File download

Assignment 3:

- Navigate to "Window Operations" page by clicking "Window Operations" button displayed on the page just opened.
- Click the buttons and verify the actions.

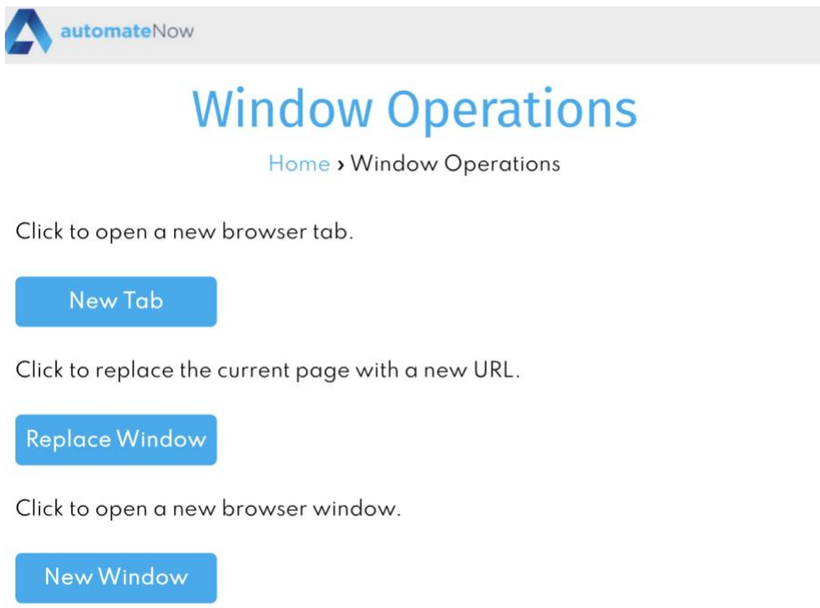

Figure 4 – Windows Operations

Assignment 4: (APIs - https://dummy.restapiexample.com)

Test case 1:

- Verify GET method with the following dummy APIs

  https://dummy.restapiexample.com/api/v1/employees

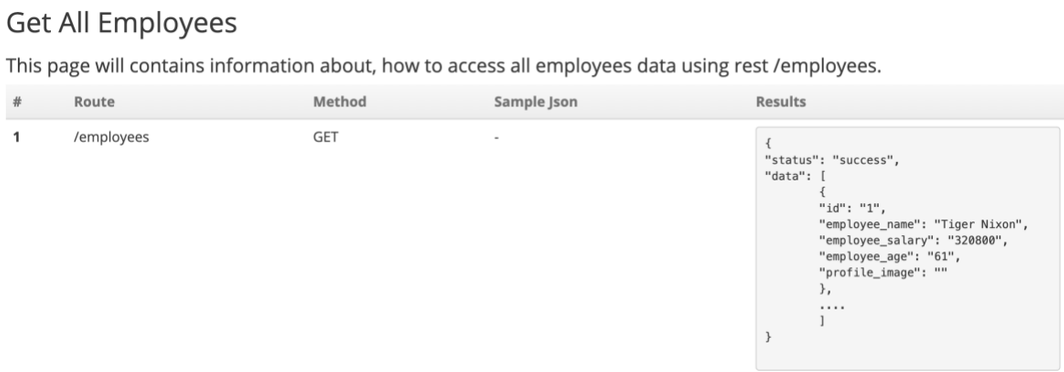

Figure 5 – GET Rest API response

  - Step 1: Perform a 'GET' request to the above API
  - Step 2: Verify that the response code is '200'.
  - Step 3: Verify that the 'employee_name' and other fields key in the response body has the value "<your observed value in the response>".

Test case 2:

- Verify POST method with the following dummy API

https://dummy.restapiexample.com/create

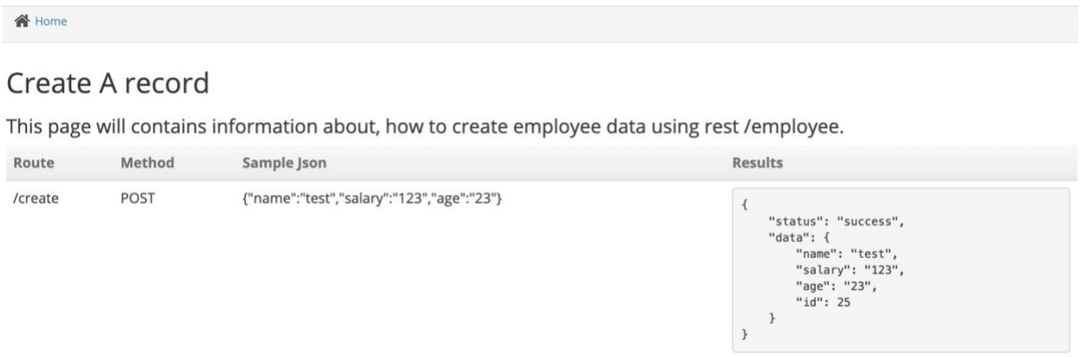

Figure 6 – POST Rest API response

- Step 1: Perform a 'POST' request to the above UTL with header and body.
- Step 2: Verify that the response code is '200'.
- Step 3: Verify that the value of the response as expected.

Test case 3:

- Verify GET method on a single item

https://dummy.restapiexample.com/api/v1/employee/1

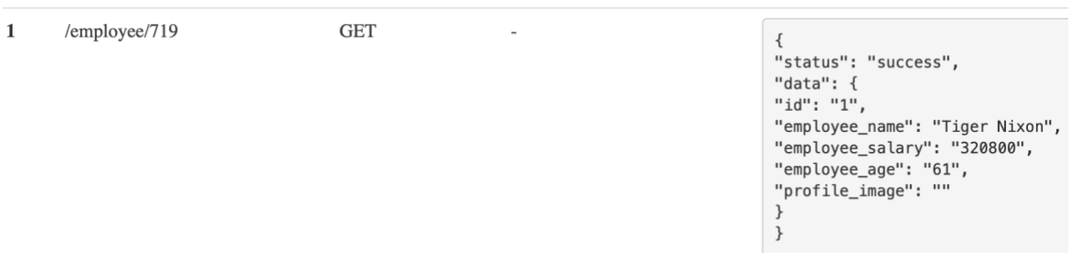

Figure 7 – GET on a single item

- Step 1: Perform a 'GET' request to the above API
- Step 2: Verify that the response code is '200'.
- Step 3: Verify that the 'employee_name' and other fileds key in the response body has the value "<your observed value in the response>".

Test case 4:

- Verify PUT method

https://dummy.restapiexample.com/api/v1/update/21

Route	Method	Sample Json	Results
/update/{id}	PUT	{"name":"test","salary":"123","age":"23"}	

Figure 8 – PUT method

- Step 1: Perform a 'PUT' request to the above API with a body
- Step 2: Verify that the response code is '200'.
- Step 3: Verify that your edited information in the response

Test case 5:

- Verify DELETE method

  https://dummy.restapiexample.com/api/v1/delete/2

#	Route	Method	Sample Json	Results
1	/delete/719	GET	-	

Figure 9 – DELETE method

- Step 1: Perform a 'GET' request to the above API
- Step 2: Verify that the response code is '200'.
- Step 3: Verify the messages responded.

4. REFERENCES

[1] Daich, G., Price, G., Ragland, B., Dawood, M. "Software Test Technologies Report." STSC, Hill Air Force Base, Utah, August 1994.

[2] https://robotframework.org/

[3] https://practice-automation.com/

[4] https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html

5. REVISION HISTORY

Revision	Date	Author(s)	Description
1.0	Dec 2023	Nguyen Thanh Quan (MEng)	Created
1.1	April 2024	Nguyen Thanh Quan (MEng)	Added Rest APIs