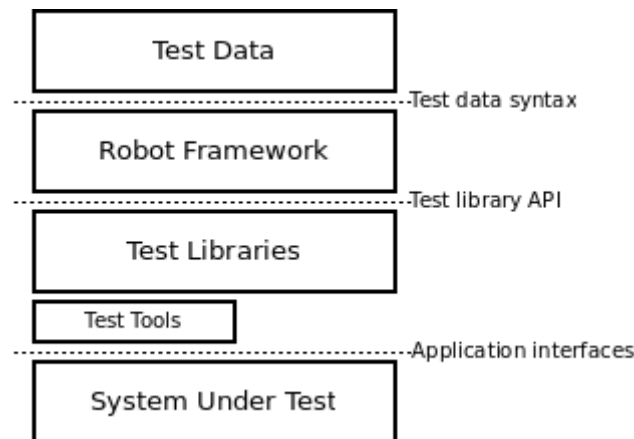


# Web automation using Robot Framework with Python and Selenium Library

## High-level architecture:

Robot Framework is a generic, application and technology independent framework. It has a highly modular architecture illustrated in the diagram below.



*Robot Framework architecture*

Here are step-by-step instructions for setting up web automation using Robot Framework with Python and Selenium Library:

## Install Python:

If you haven't already installed Python, download and install it from the official website: <https://www.python.org/downloads/>.

## Install Robot Framework:

Once Python is installed, open a command prompt and execute the following command:

pip install robotframework

```
C:\WINDOWS\system32\cmd.exe
C:\Users\DEPURANIK>pip list
Package Version
-----
pip      10.0.1
setuptools 39.1.0
You are using pip version 10.0.1, however version 24.0 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\Users\DEPURANIK>pip install robotframework
Collecting robotframework
  Downloading https://files.pythonhosted.org/packages/b5/f0/338803e955926c7826cfe6738ed99db1f76c3a86defca2c33228b12119a
/robotframework-6.1.1-py3-none-any.whl (469kB)
    100% |#####| 706kB 5.1MB/s
Installing collected packages: robotframework
Successfully installed robotframework-6.1.1
You are using pip version 10.0.1, however version 24.0 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\Users\DEPURANIK>pip list
Package Version
-----
pip      10.0.1
robotframework 6.1.1
setuptools 39.1.0
You are using pip version 10.0.1, however version 24.0 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\Users\DEPURANIK>
```

## Install Selenium Library:

SeleniumLibrary is a web testing library for Robot Framework. Install it using pip:

pip install robotframework-seleniumlibrary

## Set Up an Editor:

Choose an editor for writing Robot Framework scripts. Options include PyCharm, Eclipse, Sublime Text, Notepad++, Visual Studio, etc. For this tutorial, we'll use Notepad++.

## Download ChromeDriver:

Download the ChromeDriver executable from the official ChromeDriver website:

<https://sites.google.com/a/chromium.org/chromedriver/>.

Make sure to download the appropriate version compatible with your Chrome browser.

## Create a New Directory:

Create a new directory to store your Robot Framework test files and place a ChromeDriver executable in the Scripts folder of your Python installed path.


## Write Your Test Script:

Open Notepad++ (or your chosen editor) and create a new file with a **.robot** extension.

Write your test script using Robot Framework syntax.

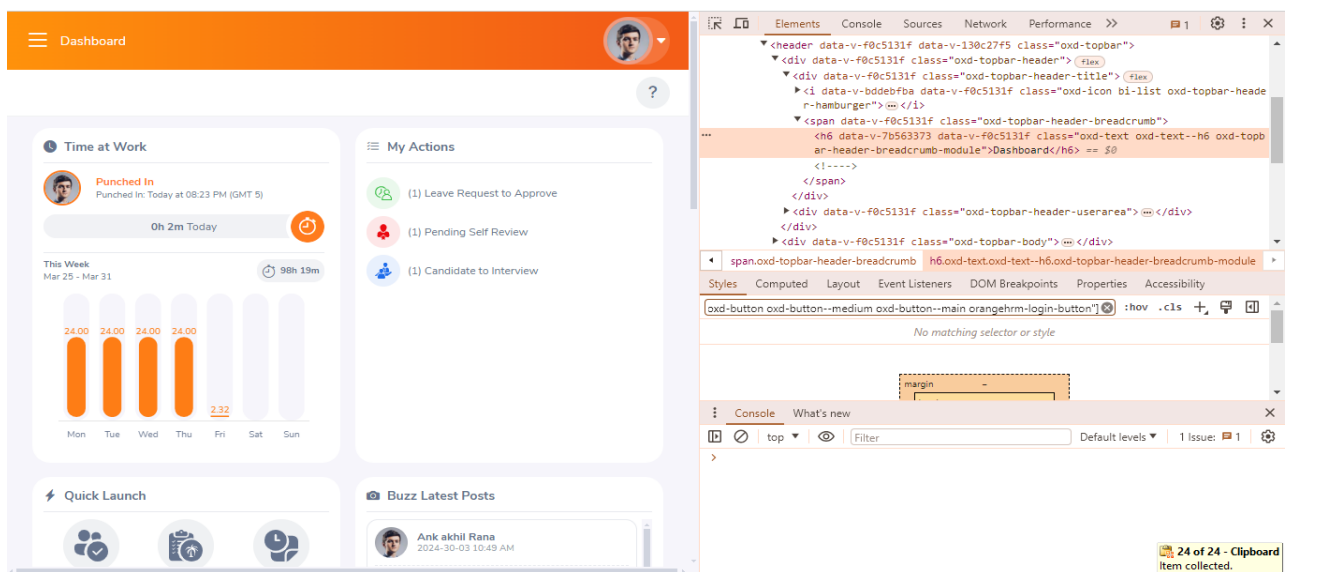
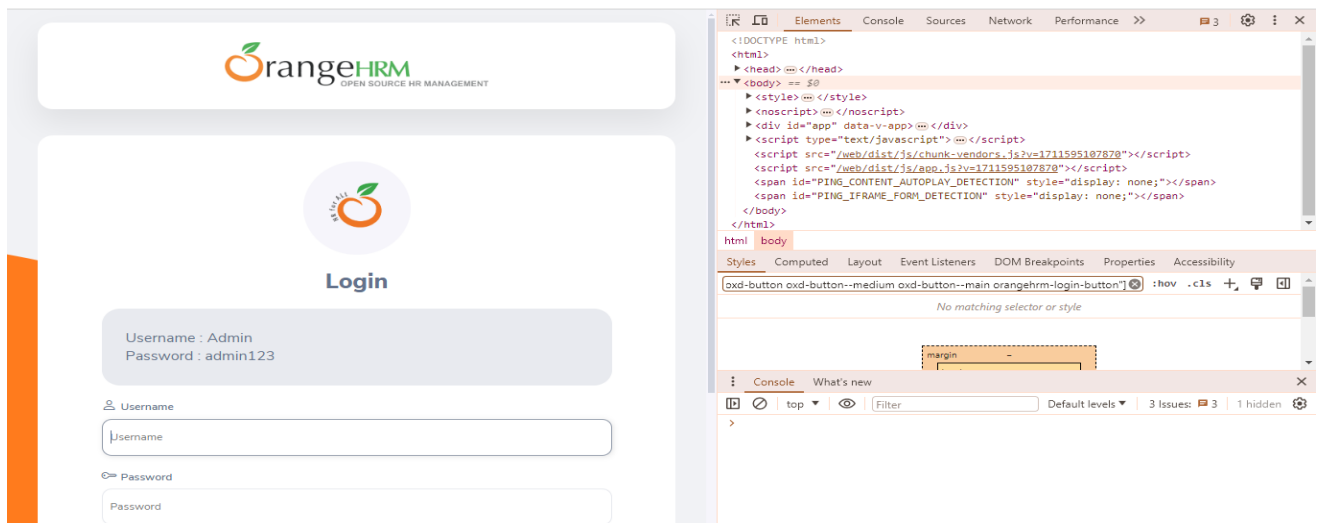
Keyword Document Reference: <https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html>

For example:



```
1  *** Settings ***
2  Library           SeleniumLibrary
3
4  *** Variables ***
5  ${BROWSER}       Chrome
6  ${URL}           https://opensource-demo.orangehrmlive.com/
7  ${USERNAME}      Admin
8  ${PASSWORD}      admin123
9
10 *** Keywords ***
11 Enter USERNAME
12     Input Text    css=input[name="username"]    ${USERNAME}
13 Enter PASSWORD
14     Input Text    css=input[name="password"]    ${PASSWORD}
15 Click on Login Button
16     Click Element  css=.orangehrm-login-button
17 Verify Dashboard for user
18     Wait Until Page Contains Element  css=h6[data-v-7b563373][data-v-f0c5131f]  timeout=10s
19     ${dashboard_header_text} =      Get Text    css=h6[data-v-7b563373][data-v-f0c5131f]
20     Should Be Equal As Strings      ${dashboard_header_text}    Dashboard
21
22 *** Test Cases ***
23 Login Test With Valid credentials
24     [Tags]         Regression
25     [Setup]        Open Browser    ${URL}    ${BROWSER}
26     Maximize Browser Window
27     Wait Until Page Contains Element  css=input[name="username"]  timeout=10s
28     Enter USERNAME
29     Enter PASSWORD
30     Click on Login Button
31     Verify Dashboard for user
32     [Teardown]    Close Browser
```

```
7  ${USERNAME}    Admin
8  ${PASSWORD}   admin123
9
10 *** Keywords ***
11 Enter USERNAME
12     Input Text    css=input[name="username"]    ${USERNAME}
13 Enter PASSWORD
14     Input Text    css=input[name="password"]    ${PASSWORD}
15 Click on Login Button
16     Click Element    css=.orangehrm-login-button
17 Verify Dashboard for user
18     Wait Until Page Contains Element    css=h6[data-v-7b563373][data-v-f0c5131f]    timeout=10s
19     ${dashboard_header_text} =    Get Text    css=h6[data-v-7b563373][data-v-f0c5131f]
20     Should Be Equal As Strings    ${dashboard_header_text}    Dashboard
21 Logout From the Application
22     Click Element    css=img[data-v-bdd6d943][alt="profile picture"]
23     Click Element    css=a[href="/web/index.php/auth/logout"]
24     Sleep    3s
25 *** Test Cases ***
26 Login and Logout Test With Valid credentials
27 [Tags]    Regression
28 [Setup]    Open Browser    ${URL}    ${BROWSER}
29 Maximize Browser Window
30 Wait Until Page Contains Element    css=input[name="username"]    timeout=10s
31 Enter USERNAME
32 Enter PASSWORD
33 Click on Login Button
34 Verify Dashboard for user
35 Logout From the Application
36 [Teardown]    Close Browser
```



In this script:

- **Sections:**
  - **Settings:** Import libraries and define settings.
  - **Variables:** Define variables such as browser, URL, username, and password.
  - **Test Cases:** Define test cases along with setup, teardown, and actual test steps.
  - **Keywords:** Define custom keywords for reusable actions.
- **Documentation:** Use the **[Documentation]** setting to provide descriptive comments about test cases or sections.
- **Tags:** Use **[Tags]** to categorize test cases for easy filtering and organization.
- **Setup and Teardown:**
  - **[Setup]** is used to perform actions before executing test cases (e.g., opening the browser).
  - **[Teardown]** is used to perform clean up actions after executing test cases (e.g., closing the browser). Even if there is any failure it make sure to execute the command in the **[Teardown]**
- **Browser Maximization:** Use **Maximize Browser Window** to maximize the browser window for better visibility during test execution.
- **Assertions:** Use **Should Be Equal As Strings** to verify that the expected text matches the actual text on the page.

For using Headless Chrome, you can specify it in the **\$(BROWSER)** variable as **"headlessChrome"**. This runs the activity in the background and the execution speed will be high. Selenium screenshots are typically available after test execution if the selenium built in functions fail to do an action and it will capture and provide the detailed reports. Screenshots are not available if the failure is because of any other assertions.

### **Running in Command Line Interface(CLI):**

Open the Command prompt in the path where your Robot scripts are placed.

To run a specific script -> robot Filename.robot

To run all the scripts (Executes sequentially) -> robot \*.robot

To run the scripts which are starting from a specific word-> robot My\*.robot

### **Running tests in parallel:**

Go to <https://pabot.org/> and click on Github link for more information.

Install pabot: `pip install -U robotframework-pabot`

### **Command-line options**

`pabot [--verbose|--testlevelsplits|--command .. --end-command|`  
`--processes num|--pabotlib|--pabotlibhost host|--pabotlibport port|`  
`--processtimeout num|`  
`--shard i/n|`  
`--artifacts extensions|--artifactsinsubfolders|`  
`--resourcefile file|--argumentfile[num] file|--suitesfrom file]`  
`[robot options] [path ...]`

It supports all Robot Framework Command line operations:

<https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#command-line-options-for-post-processing-outputs>

Example:

In command line type: `pabot --processes 2 --outputdir Report *.robot`

This command executes all the scripts in parallel using 2 processes and keep the consolidated reports in the Report folder.

# Logs and Reports:

```
C:\Windows\System32\cmd.exe
C:\Users\DBPURANIK\Desktop\RobotScripts>robot Login.robot
=====
Login
=====
Login Test With Valid credentials
DevTools listening on ws://127.0.0.1:14108/devtools/browser/66cc18fd-28a3-4014-8ef9-bb8a1b84b5c1
Login Test With Valid credentials | PASS |
=====
Login | PASS |
=====
1 test, 1 passed, 0 failed
=====
Output: C:\Users\DBPURANIK\Desktop\RobotScripts\output.xml
Log: C:\Users\DBPURANIK\Desktop\RobotScripts\log.html
Report: C:\Users\DBPURANIK\Desktop\RobotScripts\report.html
C:\Users\DBPURANIK\Desktop\RobotScripts>
```

```
C:\Windows\System32\cmd.exe
C:\Users\DBPURANIK\Desktop\RobotScripts>robot Logout.robot
=====
Logout
=====
Login and Logout Test With Valid credentials
DevTools listening on ws://127.0.0.1:15304/devtools/browser/4a6d403b-62db-4f7a-9c6b-182f3b0c22cf
Login and Logout Test With Valid credentials | PASS |
=====
Logout | PASS |
=====
1 test, 1 passed, 0 failed
=====
Output: C:\Users\DBPURANIK\Desktop\RobotScripts\output.xml
Log: C:\Users\DBPURANIK\Desktop\RobotScripts\log.html
Report: C:\Users\DBPURANIK\Desktop\RobotScripts\report.html
C:\Users\DBPURANIK\Desktop\RobotScripts>
```

← → ↺

File C:\Users\DBPURANIK\Desktop\RobotScripts\report.html

☆ 📁 📄 👤 ⋮

Generated  
20240330 12:59:24 UTC-04:00  
4 minutes 50 seconds ago

LOG

Login Report

Summary Information

Status: All tests passed

Start Time: 20240330 12:59:02.223

End Time: 20240330 12:59:24.601

Elapsed Time: 00:00:22.378

Log File: [log.html](#)

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	1	1	0	0	00:00:22	<div></div>

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Regression	1	1	0	0	00:00:22	<div></div>

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Login	1	1	0	0	00:00:22	<div></div>

Test Details

All Tags Suites Search

Suite:

Test:

Include:

Exclude:

Search Clear Help

FileC:\Users\DBPURANIK\Desktop\RobotScripts\log.html

20240330 12:59:24 UTC-04:003 minutes 31 seconds ago

REPORT

# Login Log

## Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	1	1	0	0	00:00:22	

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Regression	1	1	0	0	00:00:22	

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
Login	1	1	0	0	00:00:22	

## Test Execution Log

SUITE Login00:00:22.378

Full Name: Login

Source: C:\Users\DBPURANIK\Desktop\RobotScripts>Login.robot

Start / End / Elapsed: 20240330 12:59:02.223 / 20240330 12:59:24.601 / 00:00:22.378

Status: 1 test total, 1 passed, 0 failed, 0 skipped

TEST Login Test With Valid credentials00:00:21.742

Full Name: Login Test With Valid credentials

Tags: Regression

Start / End / Elapsed: 20240330 12:59:02.849 / 20240330 12:59:24.591 / 00:00:21.742

Status: PASS

SETUP SeleniumLibrary Open Browser \${URL}. \${BROWSER}00:00:15.538

KEYWORD SeleniumLibrary Maximize Browser Window00:00:00.155

KEYWORD SeleniumLibrary Wait Until Page Contains Element css=input[name="username"], timeout=10s00:00:00.643

KEYWORD Enter USERNAME00

KEYWORD Enter PASSWORD00

24 of 24 - ClipboardItem not Collected: Delete iter to increase available space

# Parallel Execution Pabot Reports:

```
C:\Windows\System32\cmd.exe
'pabot:' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\DBPURANIK\Desktop\RobotScripts>pip install -U robotframework-pabot
Collecting robotframework-pabot
  Downloading robotframework-pabot-2.18.0.tar.gz (46 kB)
    ----- 46.5/46.5 kB 1.2 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Installing backend dependencies ... done
  Preparing metadata (pyproject.toml) ... done
Requirement already satisfied: robotframework>=3.2 in c:\users\dbpuranik\pycharmprojects\untitled\venv\lib\site-packages
(from robotframework-pabot) (6.1.1)
Collecting robotframework_stacktrace>=0.4.1 (from robotframework-pabot)
  Downloading robotframework_stacktrace-0.4.1-py3-none-any.whl.metadata (8.4 kB)
Collecting natsort>=8.2.0 (from robotframework-pabot)
  Downloading natsort-8.4.0-py3-none-any.whl.metadata (21 kB)
Downloaded natsort-8.4.0-py3-none-any.whl (38 kB)
Downloaded robotframework_stacktrace-0.4.1-py3-none-any.whl (8.5 kB)
Building wheels for collected packages: robotframework-pabot
  Building wheel for robotframework-pabot (pyproject.toml) ... done
  Created wheel for robotframework-pabot: filename=robotframework_pabot-2.18.0-py3-none-any.whl size=47457 sha256=8d3ce3
f790f0ca4f87a8c13ae4e969afedc974d30a9a880b13b6f317293662ce
  Stored in directory: c:\users\dbpuranik\appdata\local\pip\cache\wheels\79\0e\44\96b48712053cbd6e80134a8d76ffc7cbdd466
afcidbe86d0f
Successfully built robotframework-pabot
Installing collected packages: robotframework_stacktrace, natsort, robotframework-pabot
Successfully installed natsort-8.4.0 robotframework-pabot-2.18.0 robotframework_stacktrace-0.4.1

C:\Users\DBPURANIK\Desktop\RobotScripts>
```

```
C:\Windows\System32\cmd.exe

C:\Users\DBPURANIK\Desktop\RobotScripts>pabot --processes 2 --outputdir Report *.robot
Storing .pabotsuitenames file
2024-03-30 14:31:32.321738 [PID:1045800] [0] [ID:0] EXECUTING Suites.Login
2024-03-30 14:31:32.321738 [PID:1045800] [1] [ID:1] EXECUTING Suites.Logout

DevTools listening on ws://127.0.0.1:20077/devtools/browser/e83ac652-3458-47a7-8327-efd1669b1fd9

DevTools listening on ws://127.0.0.1:20078/devtools/browser/6401f3b9-5e67-47f3-9835-026de9a924cd
2024-03-30 14:31:50.798331 [PID:1045800] [0] [ID:0] still running Suites.Login after 15.0 seconds
2024-03-30 14:31:50.910020 [PID:1045800] [1] [ID:1] still running Suites.Logout after 15.0 seconds
2024-03-30 14:32:15.879371 [PID:1045800] [0] [ID:0] PASSED Suites.Login in 33.1 seconds
2024-03-30 14:32:17.921773 [PID:1045800] [1] [ID:1] still running Suites.Logout after 35.0 seconds
2024-03-30 14:32:19.468442 [PID:1045800] [1] [ID:1] PASSED Suites.Logout in 36.4 seconds
2 tests, 2 passed, 0 failed, 0 skipped.

-----
Output: C:\Users\DBPURANIK\Desktop\RobotScripts\Report\output.xml
Log: C:\Users\DBPURANIK\Desktop\RobotScripts\Report\log.html
Report: C:\Users\DBPURANIK\Desktop\RobotScripts\Report\report.html
Total testing: 1 minute 9.50 seconds
elapsed time: 48.4 seconds

C:\Users\DBPURANIK\Desktop\RobotScripts>
C:\Users\DBPURANIK\Desktop\RobotScripts>
```

RobotScripts > Report				
Name	Date modified	Type	Size	
pabot_results	30-03-2024 14:31	File folder		
log	30-03-2024 14:32	Chrome HTML Do...	228 KB	
output	30-03-2024 14:32	Microsoft Edge H...	11 KB	
report	30-03-2024 14:32	Chrome HTML Do...	230 KB	

Suites Log

Test Statistics

Total Statistics							
All Tests	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip	
	2	2	0	0	00:01:13		
Statistics by Tag							
Regression	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip	
	2	2	0	0	00:01:13		
Statistics by Suite							
Suites Suites.Login Suites.Logout	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip	
	2	2	0	0	00:00:48		
	1	1	0	0	00:00:36		
	1	1	0	0	00:00:40		

Test Execution Log

SUITE Suites		00:00:47.909
Full Name: Suites		
Documentation: Pabot result from 2 executions.		
Start / End / Elapsed: 20240330 14:31:32.136 / 20240330 14:32:20.045 / 00:00:47.909		
Status: 2 tests total, 2 passed, 0 failed, 0 skipped		
+ SUITE Login		00:00:35.789
+ SUITE Logout		00:00:39.740

Suites Report

Summary Information

Status:	All tests passed
Documentation:	Pabot result from 2 executions.
Start Time:	20240330 14:31:32.136
End Time:	20240330 14:32:20.045
Elapsed Time:	00:00:47.909
Log File:	log.html

Test Statistics

Total Statistics		2	2	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests		2	2	0	0	0	00:01:13	
	Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip	
Regression		2	2	0	0	00:01:13		
Statistics by Suite		Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip	
Suites		2	2	0	0	00:00:48		
Suites_Login		1	1	0	0	00:00:36		
Suites_Logout		1	1	0	0	00:00:40		

Test Details

All	Tags	Suites	Search
Suite:			
Test:			
Include:			