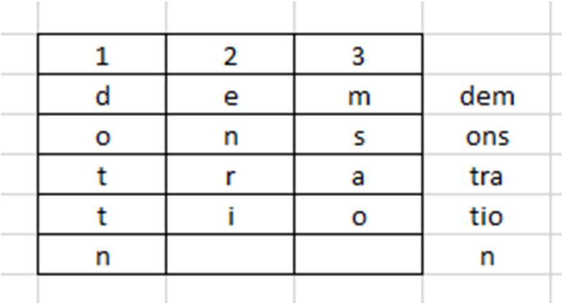
Below it is an example with the transposition cypher that I used with key value =3, for “demonstration”.



1	2	3	
d	e	m	dem
o	n	s	ons
t	r	a	tra
t	i	o	tio
n			n

Practically there are 3 columns that are created and the letters are placed row by row until we get to the last letter.

After having this arrangement, the word is joined back column by column

“dottnenrimsao” will be the result for “demonstration”.

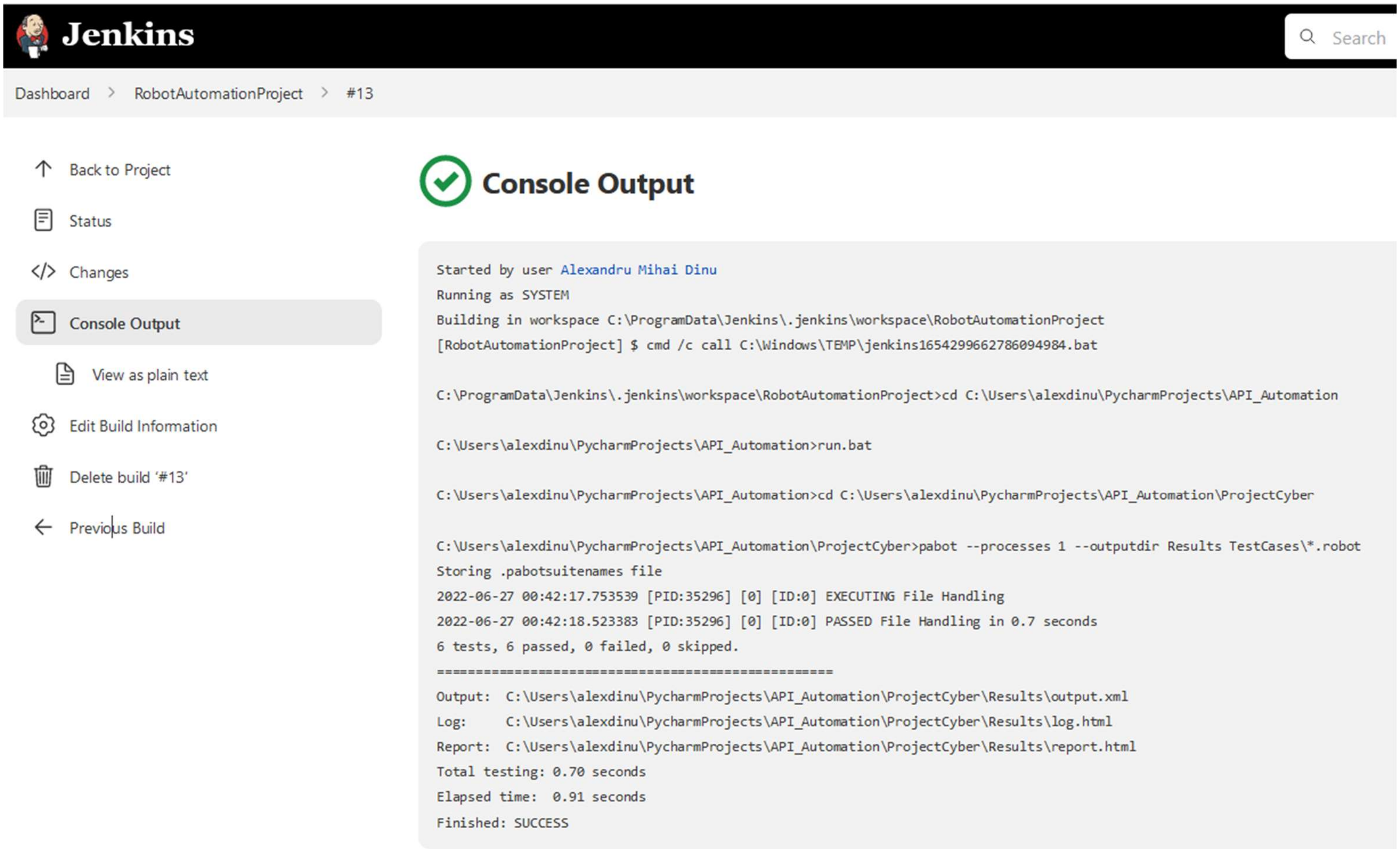
Even though I wrote 8 automated test cases, in jenkIns I ran only 6 of them, excluding the ones that handle a docx file. This is because I had some problems with docx module on my windows machine when installing it.

I first started to work on the project on a linux environment where I have completed the python code, and there I did not have issues with the docx python module.

And I switched to the windows environment because I have Jenkins in place on my windows environment, and here I hit this problem with the docx module from python.

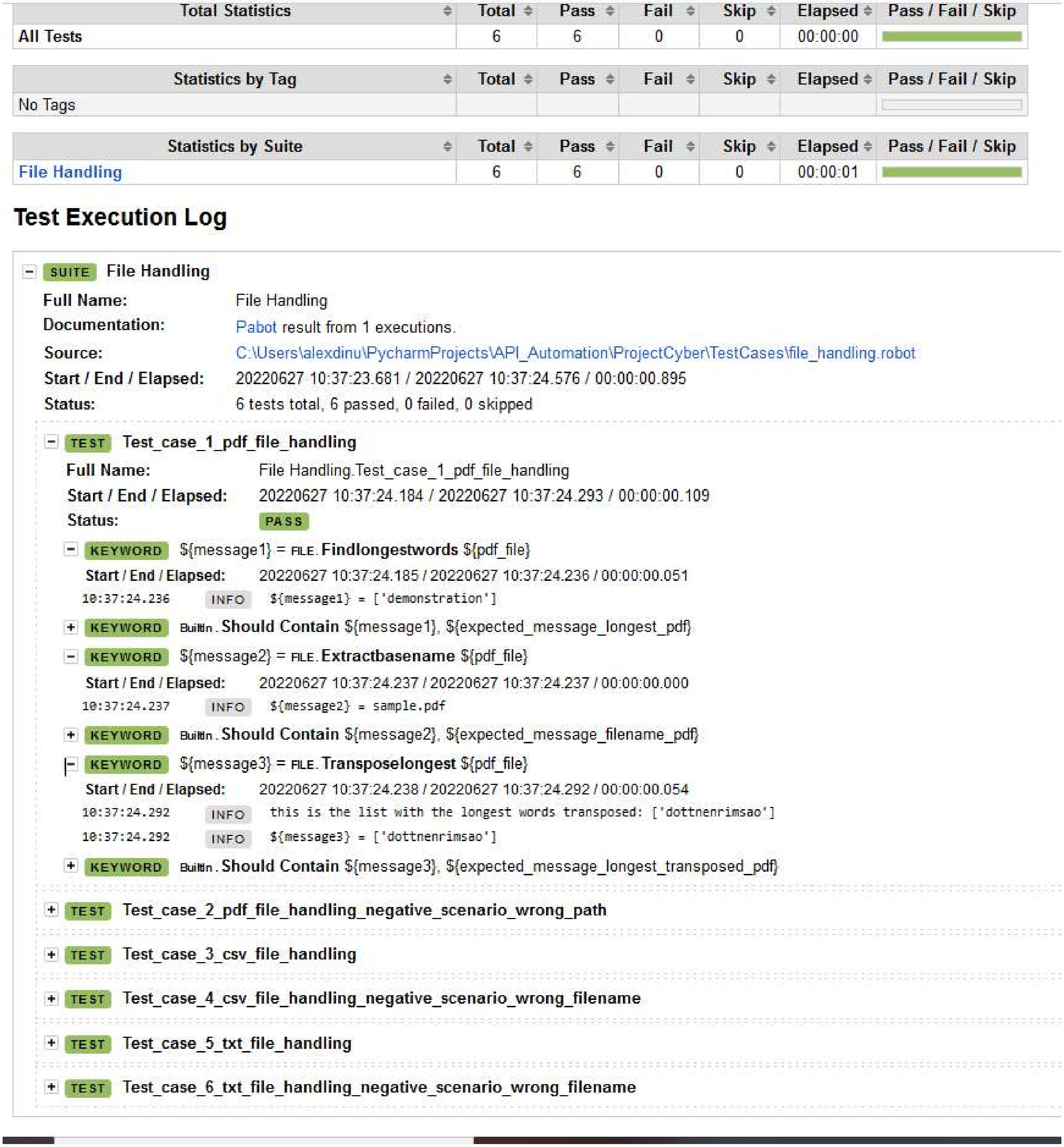
Even though I did not ran in Jenkins the tests for the docx file, I guarantee that they are written correctly.

So, here is how the Jenkins run looks successfully ran.



Console Output

```
Started by user Alexandru Mihai Dinu
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\RobotAutomationProject
[RobotAutomationProject] $ cmd /c call C:\Windows\TEMP\jenkins1654299662786094984.bat

C:\ProgramData\Jenkins\.jenkins\workspace\RobotAutomationProject>cd C:\Users\alexdinu\PycharmProjects\API_Automation

C:\Users\alexdinu\PycharmProjects\API_Automation>run.bat

C:\Users\alexdinu\PycharmProjects\API_Automation>cd C:\Users\alexdinu\PycharmProjects\API_Automation\ProjectCyber

C:\Users\alexdinu\PycharmProjects\API_Automation\ProjectCyber>pabot --processes 1 --outputdir Results TestCases\*.robot
Storing .pabotsuitenames file
2022-06-27 00:42:17.753539 [PID:35296] [0] [ID:0] EXECUTING File Handling
2022-06-27 00:42:18.523383 [PID:35296] [0] [ID:0] PASSED File Handling in 0.7 seconds
6 tests, 6 passed, 0 failed, 0 skipped.
==================================================
Output:  C:\Users\alexdinu\PycharmProjects\API_Automation\ProjectCyber\Results\output.xml
Log:     C:\Users\alexdinu\PycharmProjects\API_Automation\ProjectCyber\Results\log.html
Report:  C:\Users\alexdinu\PycharmProjects\API_Automation\ProjectCyber\Results\report.html
Total testing: 0.70 seconds
Elapsed time:  0.91 seconds
Finished: SUCCESS
```



Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	6	6	0	0	00:00:00	🟩

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
No Tags						

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
File Handling	6	6	0	0	00:00:01	🟩

Test Execution Log

− SUITE **File Handling**

Full Name: File Handling
Documentation: Pabot result from 1 executions.
Source: C:\Users\alexdinu\PycharmProjects\API_Automation\ProjectCyber\TestCases\file_handling.robot
Start / End / Elapsed: 20220627 10:37:23.681 / 20220627 10:37:24.576 / 00:00:00.895
Status: 6 tests total, 6 passed, 0 failed, 0 skipped

− TEST **Test_case_1_pdf_file_handling**

Full Name: File Handling.Test_case_1_pdf_file_handling
Start / End / Elapsed: 20220627 10:37:24.184 / 20220627 10:37:24.293 / 00:00:00.109
Status: PASS

− KEYWORD ${message1} = FILE. **Findlongestwords** ${pdf_file}

Start / End / Elapsed: 20220627 10:37:24.185 / 20220627 10:37:24.236 / 00:00:00.051

```
10:37:24.236   INFO   ${message1} = ['demonstration']
```

+ KEYWORD BuiltIn. **Should Contain** ${message1}, ${expected_message_longest_pdf}

− KEYWORD ${message2} = FILE. **Extractbasename** ${pdf_file}

Start / End / Elapsed: 20220627 10:37:24.237 / 20220627 10:37:24.237 / 00:00:00.000

```
10:37:24.237   INFO   ${message2} = sample.pdf
```

+ KEYWORD BuiltIn. **Should Contain** ${message2}, ${expected_message_filename_pdf}

− KEYWORD ${message3} = FILE. **Transposelongest** ${pdf_file}

Start / End / Elapsed: 20220627 10:37:24.238 / 20220627 10:37:24.292 / 00:00:00.054

```
10:37:24.292   INFO   this is the list with the longest words transposed: ['dottnenrimsao']
10:37:24.292   INFO   ${message3} = ['dottnenrimsao']
```

+ KEYWORD BuiltIn. **Should Contain** ${message3}, ${expected_message_longest_transposed_pdf}

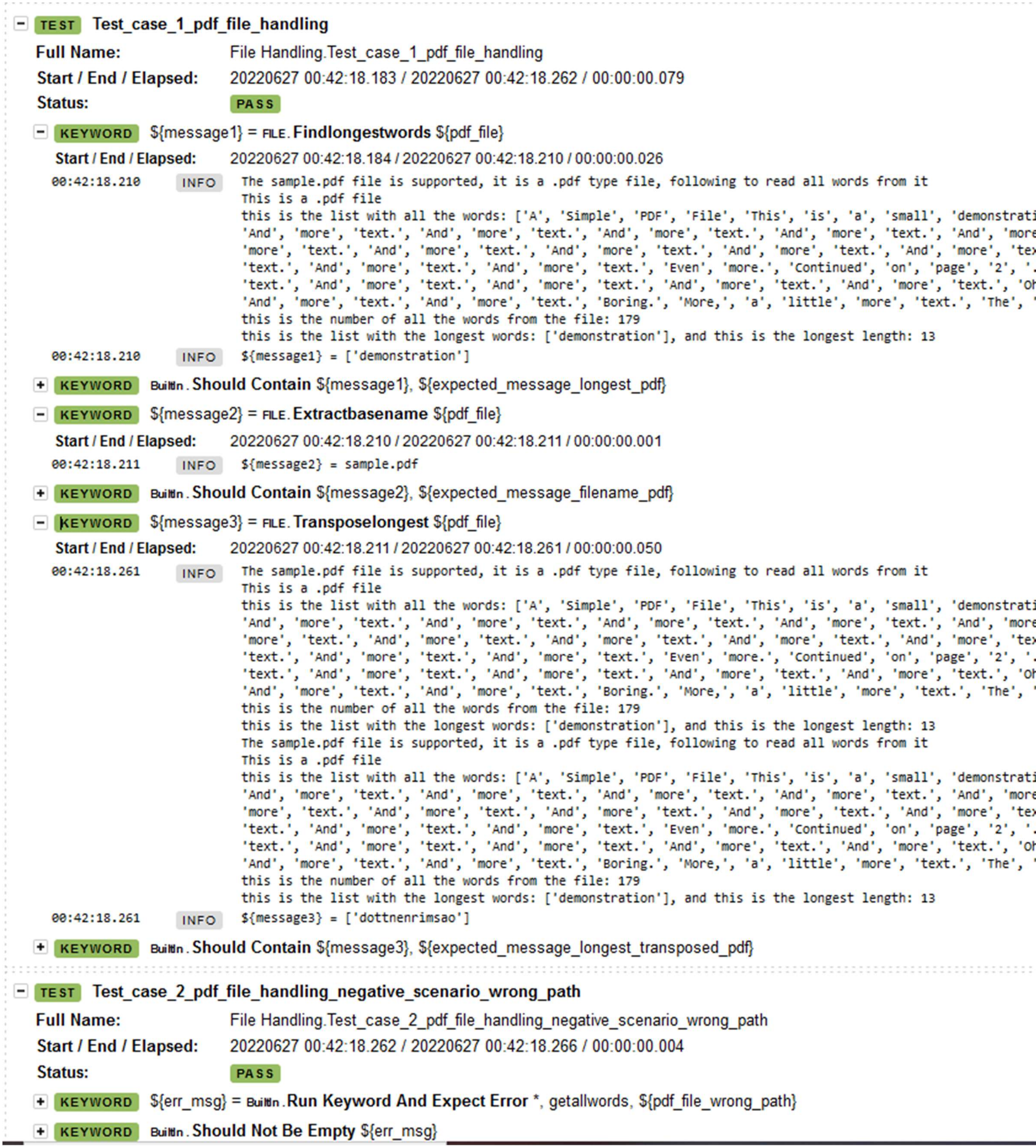
+ TEST **Test_case_2_pdf_file_handling_negative_scenario_wrong_path**

+ TEST **Test_case_3_csv_file_handling**

+ TEST **Test_case_4_csv_file_handling_negative_scenario_wrong_filename**

+ TEST **Test_case_5_txt_file_handling**

+ TEST **Test_case_6_txt_file_handling_negative_scenario_wrong_filename**



⊟ TEST **Test_case_1_pdf_file_handling**

Full Name: File Handling.Test_case_1_pdf_file_handling

Start / End / Elapsed: 20220627 00:42:18.183 / 20220627 00:42:18.262 / 00:00:00.079

Status: PASS

⊟ KEYWORD ${message1} = FILE. **Findlongestwords** ${pdf_file}

Start / End / Elapsed: 20220627 00:42:18.184 / 20220627 00:42:18.210 / 00:00:00.026

```
00:42:18.210   INFO   The sample.pdf file is supported, it is a .pdf type file, following to read all words from it
                      This is a .pdf file
                      this is the list with all the words: ['A', 'Simple', 'PDF', 'File', 'This', 'is', 'a', 'small', 'demonstrati
                      'And', 'more', 'text.', 'And', 'more', 'text.', 'And', 'more', 'text.', 'And', 'more', 'text.', 'And', 'more
                      'more', 'text.', 'And', 'more', 'text.', 'And', 'more', 'text.', 'And', 'more', 'text.', 'And', 'more', 'tex
                      'text.', 'And', 'more', 'text.', 'And', 'more', 'text.', 'Even', 'more.', 'Continued', 'on', 'page', '2', '.
                      'text.', 'And', 'more', 'text.', 'And', 'more', 'text.', 'And', 'more', 'text.', 'And', 'more', 'text.', 'Oh
                      'And', 'more', 'text.', 'And', 'more', 'text.', 'Boring.', 'More,', 'a', 'little', 'more', 'text.', 'The', '
                      this is the number of all the words from the file: 179
                      this is the list with the longest words: ['demonstration'], and this is the longest length: 13
00:42:18.210   INFO   ${message1} = ['demonstration']
```

⊞ KEYWORD BuiltIn. **Should Contain** ${message1}, ${expected_message_longest_pdf}

⊟ KEYWORD ${message2} = FILE. **Extractbasename** ${pdf_file}

Start / End / Elapsed: 20220627 00:42:18.210 / 20220627 00:42:18.211 / 00:00:00.001

```
00:42:18.211   INFO   ${message2} = sample.pdf
```

⊞ KEYWORD BuiltIn. **Should Contain** ${message2}, ${expected_message_filename_pdf}

⊟ KEYWORD ${message3} = FILE. **Transposelongest** ${pdf_file}

Start / End / Elapsed: 20220627 00:42:18.211 / 20220627 00:42:18.261 / 00:00:00.050

```
00:42:18.261   INFO   The sample.pdf file is supported, it is a .pdf type file, following to read all words from it
                      This is a .pdf file
                      this is the list with all the words: ['A', 'Simple', 'PDF', 'File', 'This', 'is', 'a', 'small', 'demonstrati
                      'And', 'more', 'text.', 'And', 'more', 'text.', 'And', 'more', 'text.', 'And', 'more', 'text.', 'And', 'more
                      'more', 'text.', 'And', 'more', 'text.', 'And', 'more', 'text.', 'And', 'more', 'text.', 'And', 'more', 'tex
                      'text.', 'And', 'more', 'text.', 'And', 'more', 'text.', 'Even', 'more.', 'Continued', 'on', 'page', '2', '.
                      'text.', 'And', 'more', 'text.', 'And', 'more', 'text.', 'And', 'more', 'text.', 'And', 'more', 'text.', 'Oh
                      'And', 'more', 'text.', 'And', 'more', 'text.', 'Boring.', 'More,', 'a', 'little', 'more', 'text.', 'The', '
                      this is the number of all the words from the file: 179
                      this is the list with the longest words: ['demonstration'], and this is the longest length: 13
                      The sample.pdf file is supported, it is a .pdf type file, following to read all words from it
                      This is a .pdf file
                      this is the list with all the words: ['A', 'Simple', 'PDF', 'File', 'This', 'is', 'a', 'small', 'demonstrati
                      'And', 'more', 'text.', 'And', 'more', 'text.', 'And', 'more', 'text.', 'And', 'more', 'text.', 'And', 'more
                      'more', 'text.', 'And', 'more', 'text.', 'And', 'more', 'text.', 'And', 'more', 'text.', 'And', 'more', 'tex
                      'text.', 'And', 'more', 'text.', 'And', 'more', 'text.', 'Even', 'more.', 'Continued', 'on', 'page', '2', '.
                      'text.', 'And', 'more', 'text.', 'And', 'more', 'text.', 'And', 'more', 'text.', 'And', 'more', 'text.', 'Oh
                      'And', 'more', 'text.', 'And', 'more', 'text.', 'Boring.', 'More,', 'a', 'little', 'more', 'text.', 'The', '
                      this is the number of all the words from the file: 179
                      this is the list with the longest words: ['demonstration'], and this is the longest length: 13
00:42:18.261   INFO   ${message3} = ['dottnenrimsao']
```

⊞ KEYWORD BuiltIn. **Should Contain** ${message3}, ${expected_message_longest_transposed_pdf}

⊟ TEST **Test_case_2_pdf_file_handling_negative_scenario_wrong_path**

Full Name: File Handling.Test_case_2_pdf_file_handling_negative_scenario_wrong_path

Start / End / Elapsed: 20220627 00:42:18.262 / 20220627 00:42:18.266 / 00:00:00.004

Status: PASS

⊞ KEYWORD ${err_msg} = BuiltIn. **Run Keyword And Expect Error** *, getallwords, ${pdf_file_wrong_path}

⊞ KEYWORD BuiltIn. **Should Not Be Empty** ${err_msg}

File Handling Log

Generated
20220627 10:37:24 UTC+03:00
31 seconds ago

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	6	6	0	0	00:00:00	🟩

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
No Tags						

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
File Handling	6	6	0	0	00:00:01	🟩

Test Execution Log

− SUITE **File Handling**

Full Name: File Handling

Documentation: Pabot result from 1 executions.

Source: C:\Users\alexdinu\PycharmProjects\API_Automation\ProjectCyber\TestCases\file_handling.robot

Start / End / Elapsed: 20220627 10:37:23.681 / 20220627 10:37:24.576 / 00:00:00.895

Status: 6 tests total, 6 passed, 0 failed, 0 skipped

+ TEST **Test_case_1_pdf_file_handling**

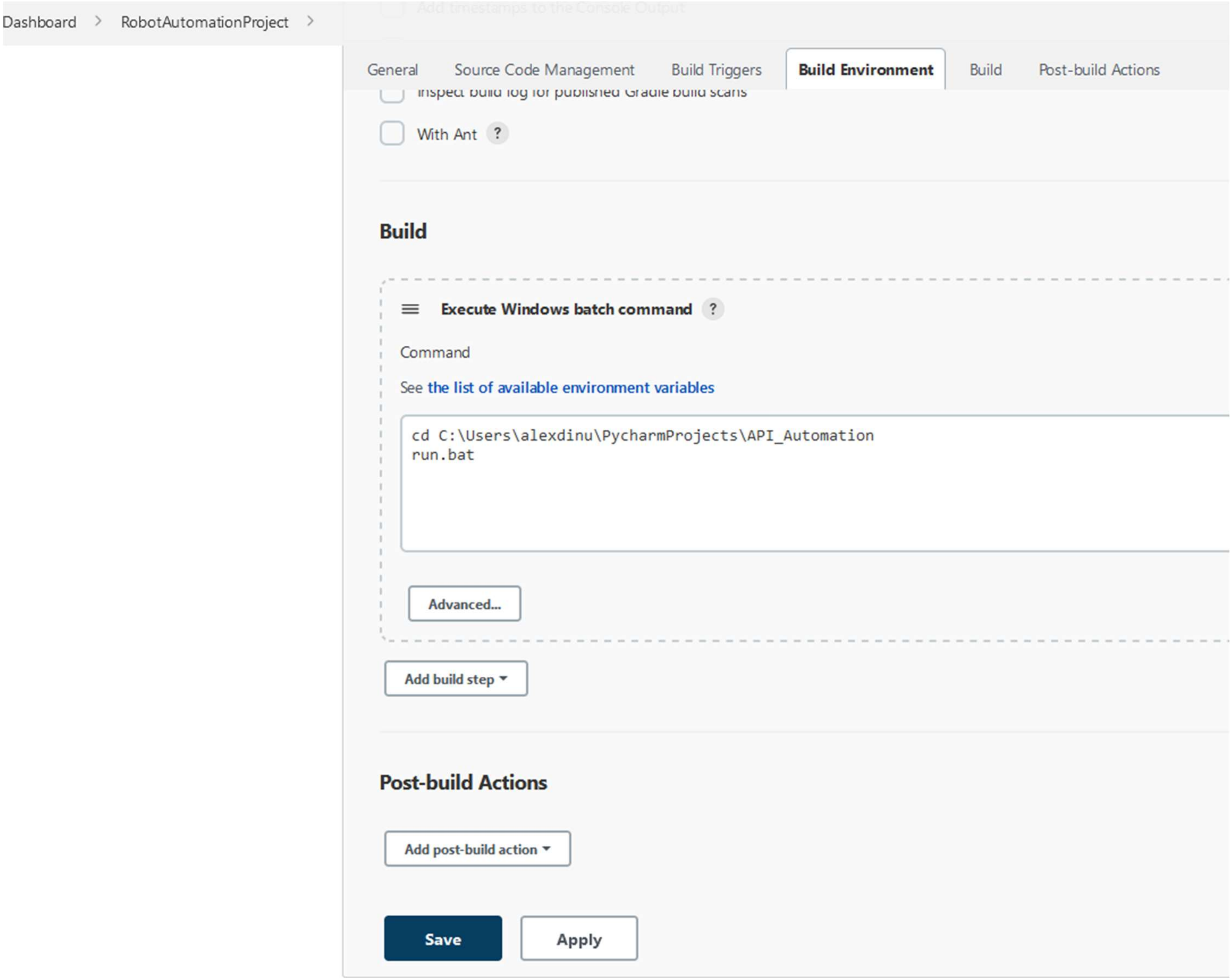
+ TEST **Test_case_2_pdf_file_handling_negative_scenario_wrong_path**

+ TEST **Test_case_3_csv_file_handling**

+ TEST **Test_case_4_csv_file_handling_negative_scenario_wrong_filename**

+ TEST **Test_case_5_txt_file_handling**

+ TEST **Test_case_6_txt_file_handling_negative_scenario_wrong_filename**



Dashboard > RobotAutomationProject >

□ Add timestamps to the Console Output

General | Source Code Management | Build Triggers | Build Environment | Build | Post-build Actions

□ Inspect build log for published Gradle build scans

Build

Execute Windows batch command ?

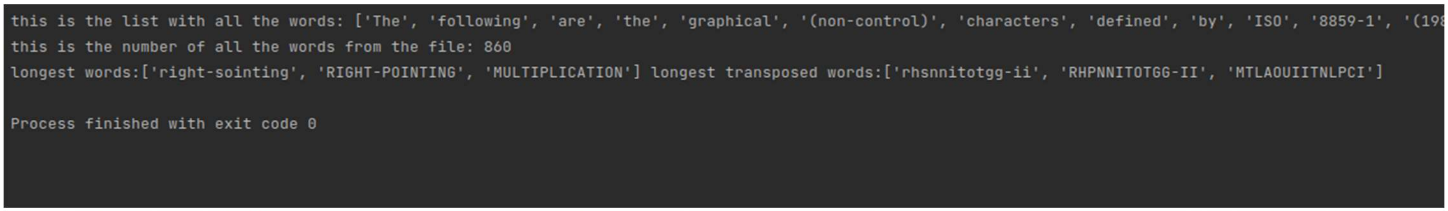
Command

See the list of available environment variables

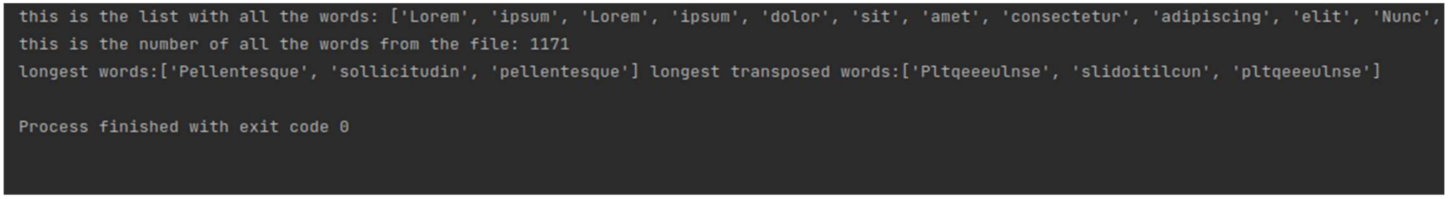
```
cd C:\Users\alexdinu\PycharmProjects\API_Automation
run.bat
```

Post-build Actions

Below are some examples of finding the longest word, where multiple words have the same length

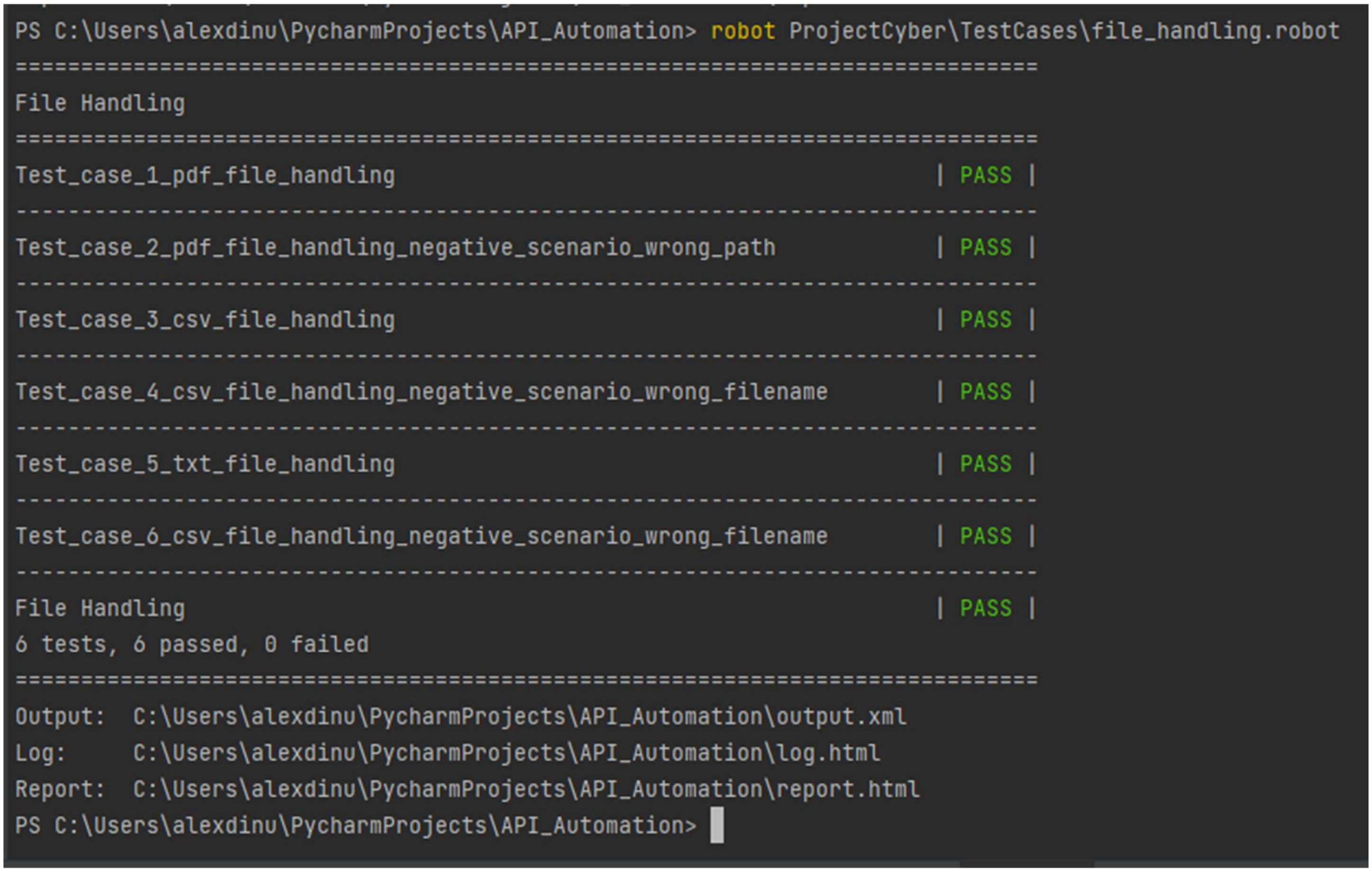


```
this is the list with all the words: ['The', 'following', 'are', 'the', 'graphical', '(non-control)', 'characters', 'defined', 'by', 'ISO', '8859-1', '(19
this is the number of all the words from the file: 860
longest words:['right-sointing', 'RIGHT-POINTING', 'MULTIPLICATION'] longest transposed words:['rhsnnitotgg-ii', 'RHPNNITOTGG-II', 'MTLAOUIITNLPCI']

Process finished with exit code 0
```



```
this is the list with all the words: ['Lorem', 'ipsum', 'Lorem', 'ipsum', 'dolor', 'sit', 'amet', 'consectetur', 'adipiscing', 'elit', 'Nunc',
this is the number of all the words from the file: 1171
longest words:['Pellentesque', 'sollicitudin', 'pellentesque'] longest transposed words:['Pltqeeeulnse', 'slidoitilcun', 'pltqeeeulnse']

Process finished with exit code 0
```

We can also ran the robot test from the pycharm terminal.



```
PS C:\Users\alexdinu\PycharmProjects\API_Automation> robot ProjectCyber\TestCases\file_handling.robot
==============================================================================
File Handling
==============================================================================
Test_case_1_pdf_file_handling                                         | PASS |
------------------------------------------------------------------------------
Test_case_2_pdf_file_handling_negative_scenario_wrong_path            | PASS |
------------------------------------------------------------------------------
Test_case_3_csv_file_handling                                         | PASS |
------------------------------------------------------------------------------
Test_case_4_csv_file_handling_negative_scenario_wrong_filename        | PASS |
------------------------------------------------------------------------------
Test_case_5_txt_file_handling                                         | PASS |
------------------------------------------------------------------------------
Test_case_6_csv_file_handling_negative_scenario_wrong_filename        | PASS |
------------------------------------------------------------------------------
File Handling                                                         | PASS |
6 tests, 6 passed, 0 failed
==============================================================================
Output:  C:\Users\alexdinu\PycharmProjects\API_Automation\output.xml
Log:     C:\Users\alexdinu\PycharmProjects\API_Automation\log.html
Report:  C:\Users\alexdinu\PycharmProjects\API_Automation\report.html
PS C:\Users\alexdinu\PycharmProjects\API_Automation>
```

I wanted to do more on this project but I focused to make each section of the challenge: the python methods, the robot test cases, jenkins run.

The methods without doubt can be improved, especially on the part of retrieving the words regarding the punctuation for example. I did not have a large amount of time to test various scenarios for punctuation, in order to improve the python methods, I focused on the text examples that I have used in my sample files.