

**SAM**

**QA Candidate Exercises**

# Copyright Information

©2018. SecureAuth® is a registered trademark of SecureAuth Corporation. SecureAuth’s IdP software, appliances, and other products and solutions are copyrighted products of SecureAuth Corporation.

Version 1.0

October 2017

For information on support for this module, contact your SecureAuth support or sales representative:

Email: [support@secureauth.com](mailto:inside-sales@secureauth.com)

inside-sales@secureauth.com

Phone: +1.949.777.6959 or +1-866- 859-1526

[Website:](https://www.secureauth.com/Contact-Us.aspx) <https://www.secureauth.com/support>

<https://www.secureauth.com/contact>



1. **Exercises**

**1.- Scripting: Word Frequency**

Using the programming language of your choice write a program that, provided a text (You can assume the text is provided as a parameter as a string), it counts

1) The number of words

2) The frequency of each of the word

3) The 5 more frequent words

(Words are considered to be case-insensitive. Discards words that appear only 1 time)

Example:

Example: with input 'This is a line\nAnd this is other' the program should produce the following output:

1. words = 8
2. Frequency: 'this' (2), 'is' (2)
3. top-5 = this, is

**2.- Test Case Creation**

For the code written for exercise 1, identify the set of testcases that you would use to ensure the code produced the desired results, classifying them according to their priority. Also describe which tests could be produced if you wanted to test for the robustness of that code.

**3.- Selenium Example**

The following is a piece of a Selenium test case that attempts the following:

1. Perform a search on google,

2. click on the last result of the first page of results

1 public class SomeGoogleSearch {

2 static WebDriver driver;

3

4 public static void main(String[] args) {

5 driver = new FirefoxDriver();

6 driver.get("http://www.google.com/");

7

8 boolean result;

9 try {

10 result = this.clickLastResult();

11 } catch(Exception e) {

12 e.printStackTrace();

13 result = false;

14 } finally {

15 /\* driver.close(); \*/

16 }

17

18 System.out.println("Test " + (result? "pased." : "failed."));

19 if (!result) {

20 System.exit(1);

21 }

22 }

23

24 private static boolean clickLastResult() {

25 //type search query

26 driver.findElement(By.name("q")).sendKeys("qa automation\n");

27

28 // click search

29 driver.findElement(By.name("btnG")).click();

30

31 // Wait 1 second for search to complete

32 Thread.sleep(10000)

33

34 // get the link of the last search result and click it

35 ArrayLIst links = driver.findElements(By.tagName("a"));

36 driver.click(links.get(links.size() -1)));

37

38 return true;

39 }

40 }

For the above code:

- identify as many issues as you can.

- identify any enhancements/improvements that you could make.

**4.- Real-life situations**: Think about the following scenarios, so you can discuss with the examiner during the interview (you don't need to write anything, just get ready for the conversation).

* **Scenario 1**: You're working on the QA team for a product. It's been 8 months since the project started and there is only 1 month to hit the project release date. All features/functionalities have been tested independently and have been already integrated into the product codebase. The QA team is focused on running the set of testcases for a full regression and the dev team is focused on fixing the defects that get reported. However, it seems that for each defect fixed a new defect gets opened, and/or an already closed defect gets re-opened. You're concerned about this situation and you want to solve the problem.

. What specific actions would you take?

. What would you say is the root cause of the problem?

. How would you solve them as a QA engineer?

. How would you solve them if you were the head of engineering (both dev and QA report to you)

* **Scenario 2**: You've just joined a QA team that’s been working on the same project for about 2 years. The project is 2 weeks away from hitting the release date and the team is focused on running a full product regression, which includes about 500 testcases. The team is composed of 5 people (including yourself) with different seniority levels and skills, but on average, each QA engineer is able to run 10 tests per day.

. What actions would you recommend to the team, so they can get the regression results available on time?

. What would be a feasible plan for the team to act? (i.e.: Who does what?)

. How would you present this plan to the QA manager / executives in the company? What kind of information would you use?

* **Scenario 3**: You join a QA team that owns a test suite of 1000 test cases, and they have around 80% automation coverage. You approach their framework and find out it’s fragile and takes a lot of maintenance each sprint to keep working and delivering reliable results. Release date is only two months away.

. What would be your position in regards of that framework?

. What actions would you take in order to improve the automation robustness and reliability? How would you communicate that to your superior?

# Answers of Jonathan Fernandez

Points 1 y 2 (TestScriptingWordFrequency.cs) are delivered in a zip with solution Exercises\_SecureAut.

2)

|  |  |  |
| --- | --- | --- |
| **Tc Name** | **Priority** | **Summary** |
| CountWords | Alta | The number of words |
| Frequency | Alta | The frequency of each of the word |
| CountWordsFormat | Baja | Message is Similary words = 8 |
| FrequencyFormat | Baja | Message is Similary Frequency: 'this' (2), 'is' (2) |
| FiveMoreFrequentWords | Alta | Validate the visualization of the 5 most frequent words. |
| FiveMoreFrequentWordsFormat | Baja | Message Similary top-5 = this, is |
| IgnoreWords | Alta | Discards words that appear only 1 time |
| WordsCaseInsensitive | Alta | Validate that words are not case-sensitive. |

3) **Incidents found**

001 - Driver close commented, it is important to close the driver at the end

14 } finally {

15 /\* driver.close(); \*/

16 }

002 – He was expected to wait 1 second, and wait 10

// Wait 1 second for search to complete

32 Thread.sleep(10000)

003 – The following line will give exception, something similar to driver.findelement (element) .click should be used

driver.click(links.get(links.size() -1)));

**Recommended improvements**

* Apply page object model
* Create Setup method to instantiate the driver
* Apply unit test model as Junit
* Avoid thread.sleep , use explicit waits