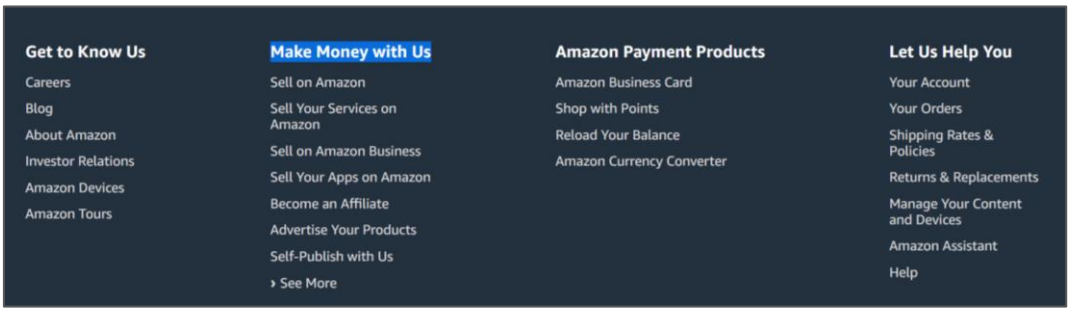
GUI Testing – Exercises

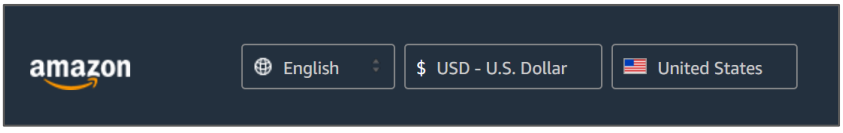
1. Install Katalon Recorder as a browser extension for Chrome. Then, create a new test suite and record your first test case: you should open the website of Amazon (https://www.amazon.com/) and check that the title of the page is equal to "*Amazon.com: Online Shopping for Electronics, Apparel, Computers, Books, DVDs & more*". Note that the expected value must be entered in the "Target" field.

2. Let us create a new test: open the website of Amazon and check that the text "*Today's Deals*" appear somewhere within the main page (command `assertTextPresent`).

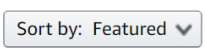
3. Open the website of Amazon and check that the page includes the text "*Make Money with Us*" as the header of the second column of links in the footer (command `assertText`).



4. Open the website of Amazon and use the dropdown menu at the bottom of the page to select the Euro currency. Once selected, check that the selected text in the menu is equal to "*EUR - Euro*".



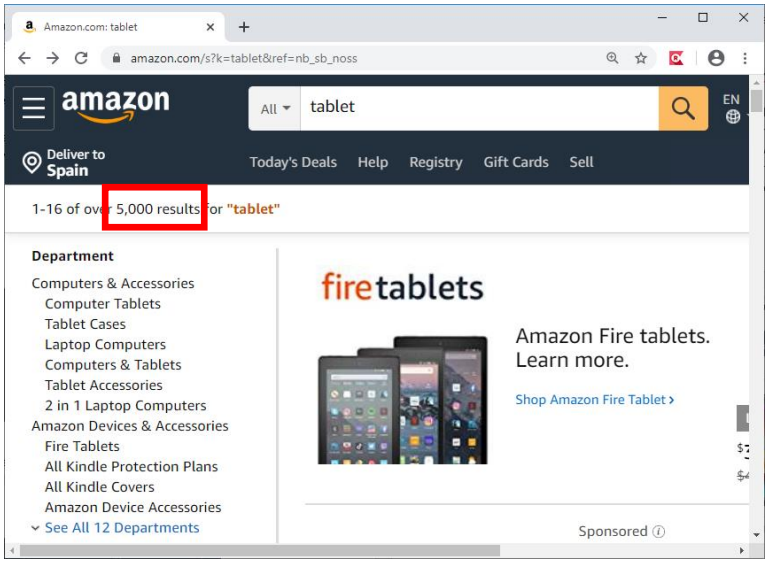
5. Open the website of Amazon and perform a search for products using the keyword "*tablet*". Then, check that the default **value** of the ordering criterion selector is "*relevanceblender*" (command `assertValue`).



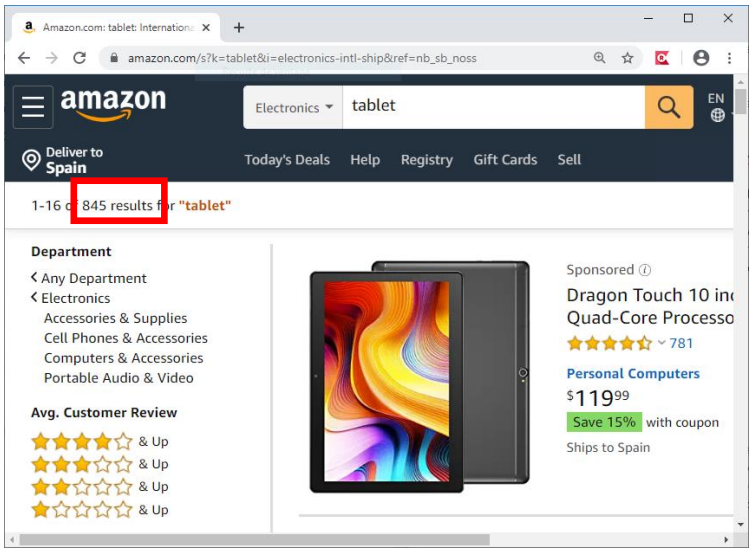
6. Open the website of Amazon and perform a search for products using the keyword "*tablet*". Then, check that the title of the page is "*Amazon.com: tablet*".

7. Repeat the previous test, but this time use a variable "keyword" to store the search keyword before using it. To create the variable you must use the command `Store` with Target="tablet" and Value="keyword". After this, you should be able to access the value of the variable using the syntax ${variable_name}.

8. Let's perform our first metamorphic tests. Perform a search for "*tablet*" and store the number of results in a variable with name "result" (command `storeText`). Next, perform a search for the same keyword restricting it to the "*Electronics*" department. Finally, check that the number of results of the latter search (filtered) is not equal to the number of results obtained in the former search.



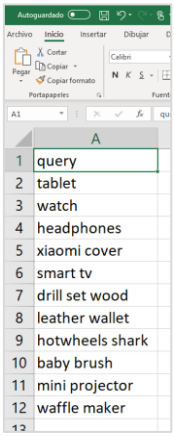
Source test case



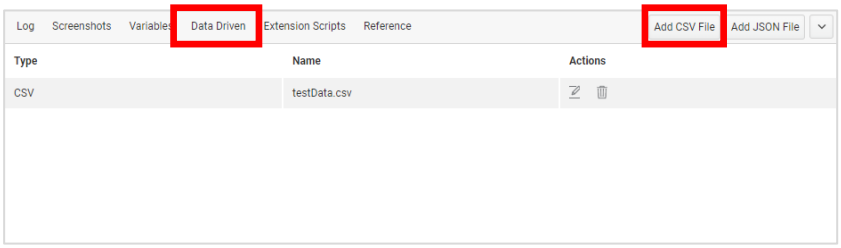
Follow-up test case

9. Perform two searches with the same keyword using a different sorting criterion on each of them, for example, "*Price: Low to High*", and "*Price: High to Low*". The number of results of both searches should be equal.

10. Let's generalize the previous test using a data-driven approach. Create a CSV file named "testData.csv" including a single column with 10 random search queries. The header of the column should be "query". Next, upload the file to Katalon Recorder using the button "*Add CSV file*" of the "*Data Driven*" tab in the menu below. Once loaded, the variable "query" should be accessible from the test case designer.



	A
1	query
2	tablet
3	watch
4	headphones
5	xiaomi cover
6	smart tv
7	drill set wood
8	leather wallet
9	hotwheels shark
10	baby brush
11	mini projector
12	waffle maker



11. Create a new test suite and create some test cases for the application https://katalon-demo-cura.herokuapp.com/

Advanced exercises

It is time to code. Download the Java (Eclipse) project at https://github.com/ssegura/GUITesting. The project is configured to use the libraries of Junit 5 and the extension for Selenium developed by bonigarcia: https://github.com/bonigarcia/selenium-jupiter. This extension makes straightforward to implement Junit 5 test cases using Selenium on multiple browsers, and even with docker containers. Have a look to the documentation at https://bonigarcia.github.io/selenium-jupiter/

1. Complete the TODO tasks in the test file *UniLu.java* and run the test (as Gradel Tests).

2. Create a new test case *Amazon.java* and implement all the test cases developed in the exercises 1-10. You may find helpful the export feature of Katalon Recorder.