Testing the page Musixmatch

**Test Plan**

**Product Under Test:**

the webpage <https://www.musixmatch.com/es> will be the product tested ,

the business goal of this product is to bring music to the customers straight from the music industry. having the biggest music catalog of lyrics in the world and letting publishers with reports of summarizing data on third party platforms to pay royalties directly to the right owners.

**Business Case:**

Why are we doing this test?

we're testing the musixmatch page to verify the quality of the following areas: “login”, “sign-in”, “music share”, “and ability to delete accounts”.

What are the benefits?

The benefits of testing these elements is to test the quality of the main workflows of the page to ensure its quality.

What are the risks of not testing?

If the main workflow of the client breaks it will cause a loss of clients creating a significant loss in the company's profit.

**Test Objectives:**

goals:

The goal of this test is to ensure the quality of the main user flow across the webpage.

First the user should be able to register if the account is not created.  
Second, once the account is registered, the user should be able to acces its profile.

Third, the user should be able to show his friends the music he likes.

Last, the user should be able to delete his account if he likes so.

Extra, test the API by making several GET requests inside the page and on Postman.

Hypothesis- the main user flow should be able to work correctly without any crash, even if we test the API by making several GET requests.

**Participants:**

1 QA tester

**Equipment:**

One Macbook

**Location and Dates:**

Monterrey Mexico from May 19 till May 20

**Cost of test:**

The respective cost of the test for the 2 days would be around $1000 pesos for the labor force in the process.

**Test Strategy**

The test strategy will be to test each component functionality individually, and for the ones who need a login first will be run with the login and its respective test.

The main user flow will be run with the extension of Selenium browser, which is a Rec and play testing method.

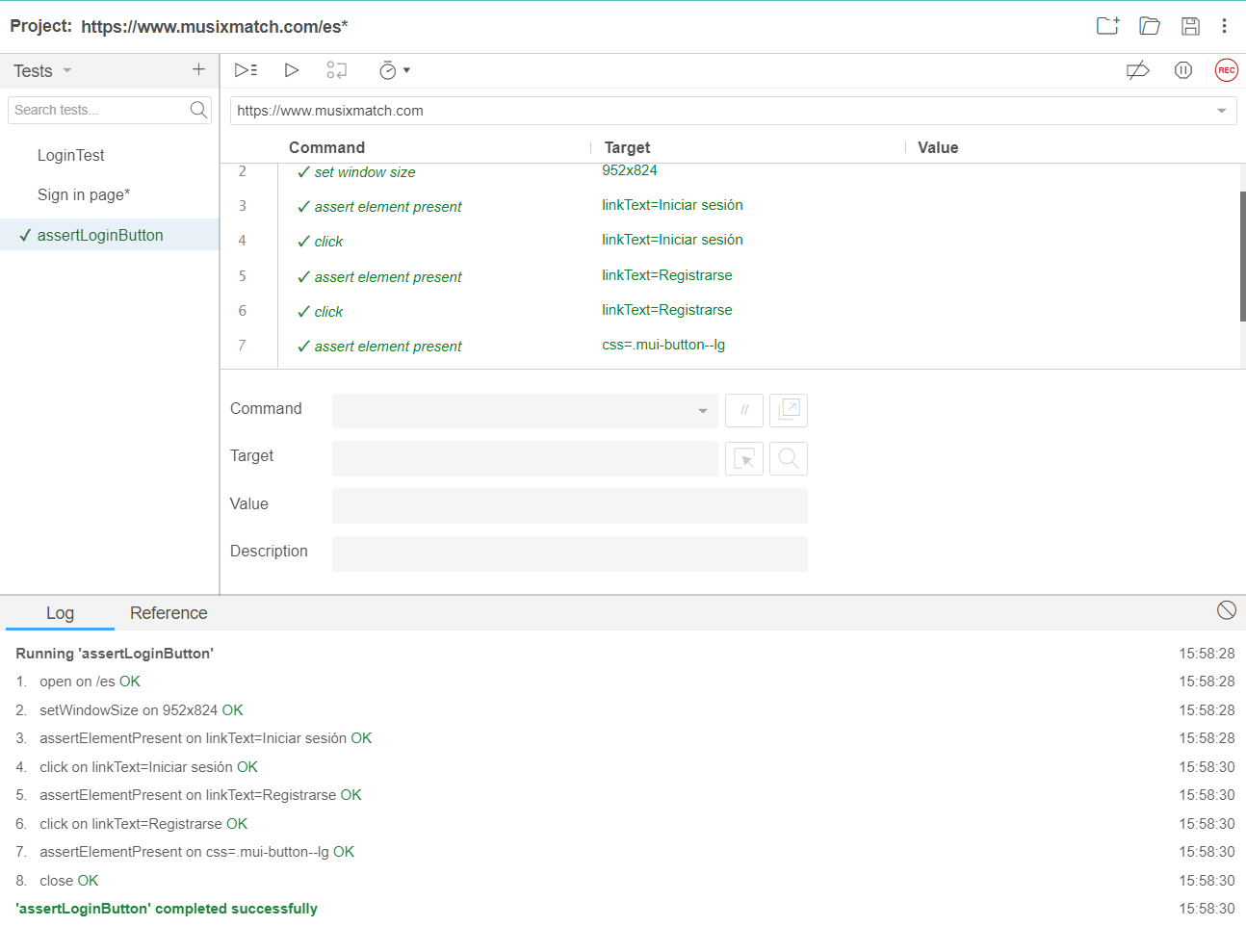
The integration features will be tested with Postman and will be by getting several Get requests to see if the API gets broken or not.

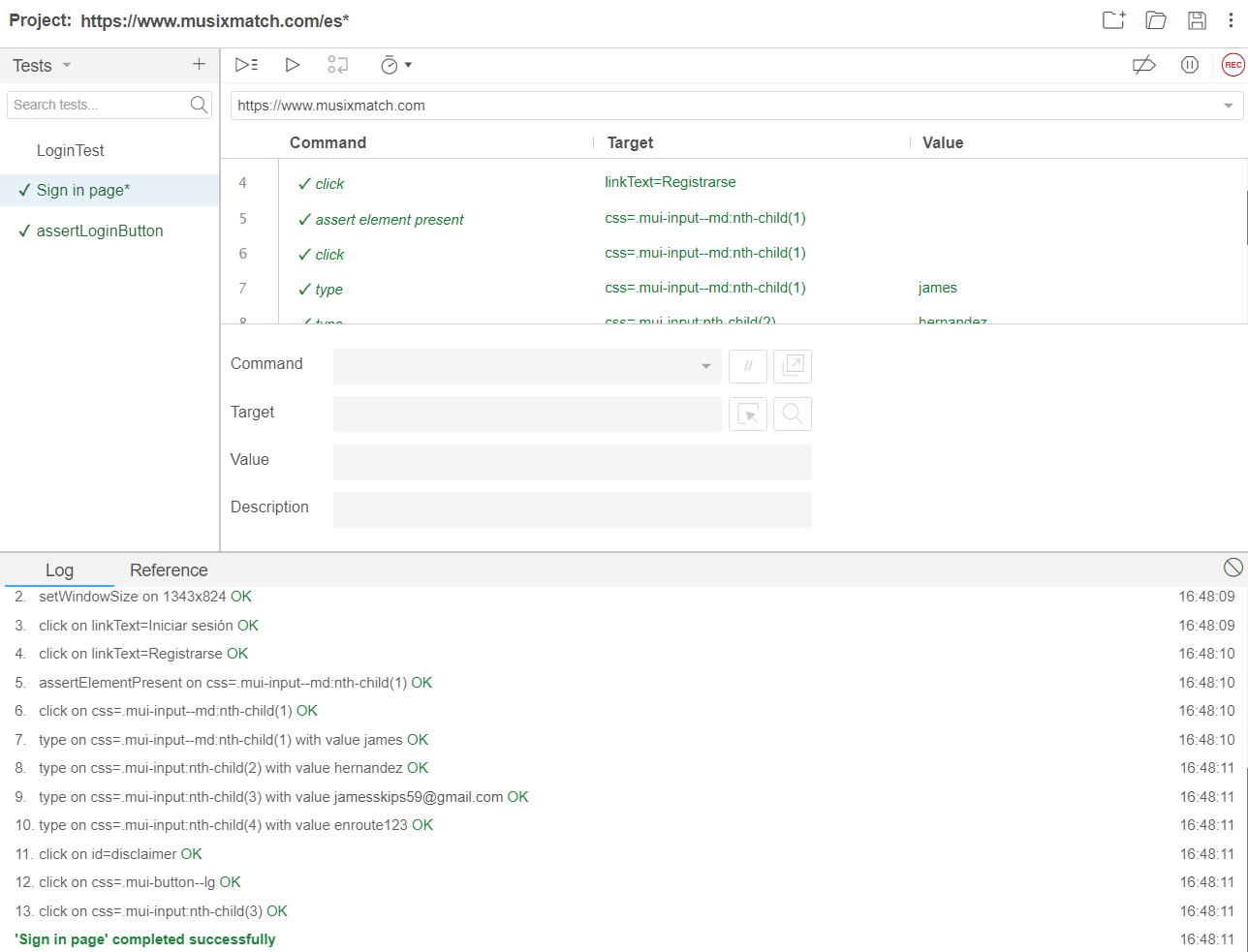
**Test Matrix**

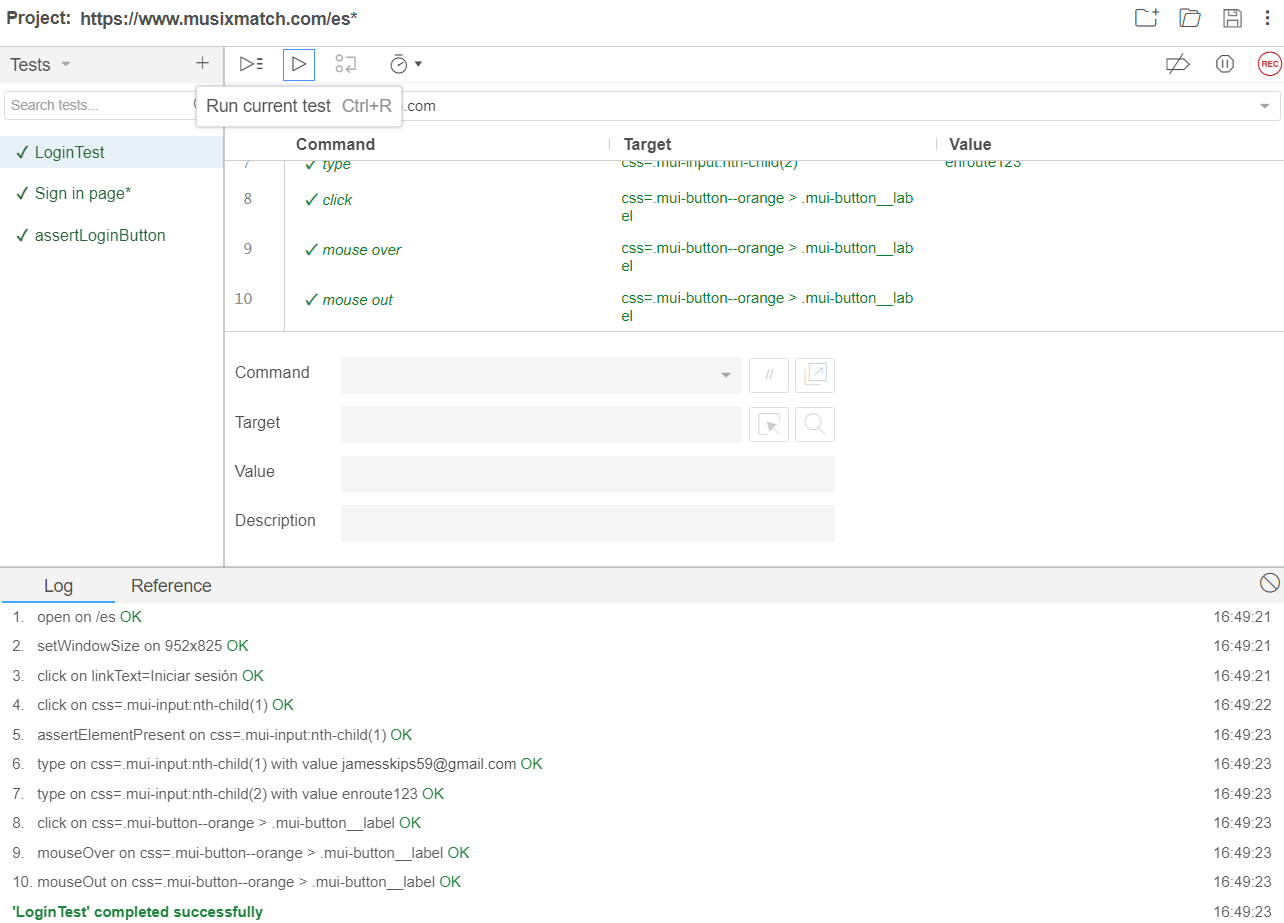
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test |  |  |  |  |
|  | Asserts | flow work | favorites | delete works |
| Registration | x | x |  |  |
| Login page | x | x |  |  |
| Music share | x |  | x |  |
| Delete Account | failed |  |  | failed |

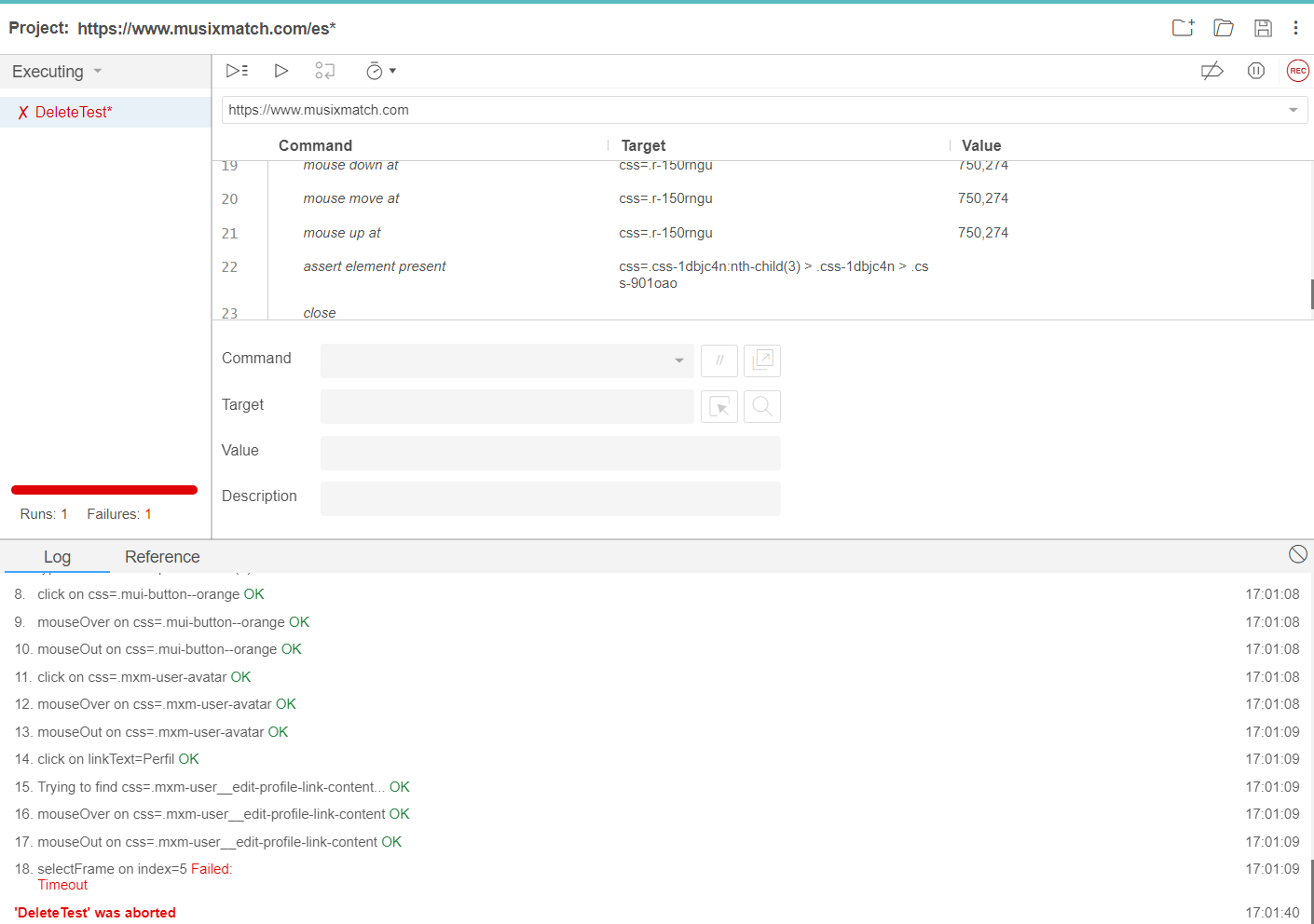
**Resultados**

**log in test**

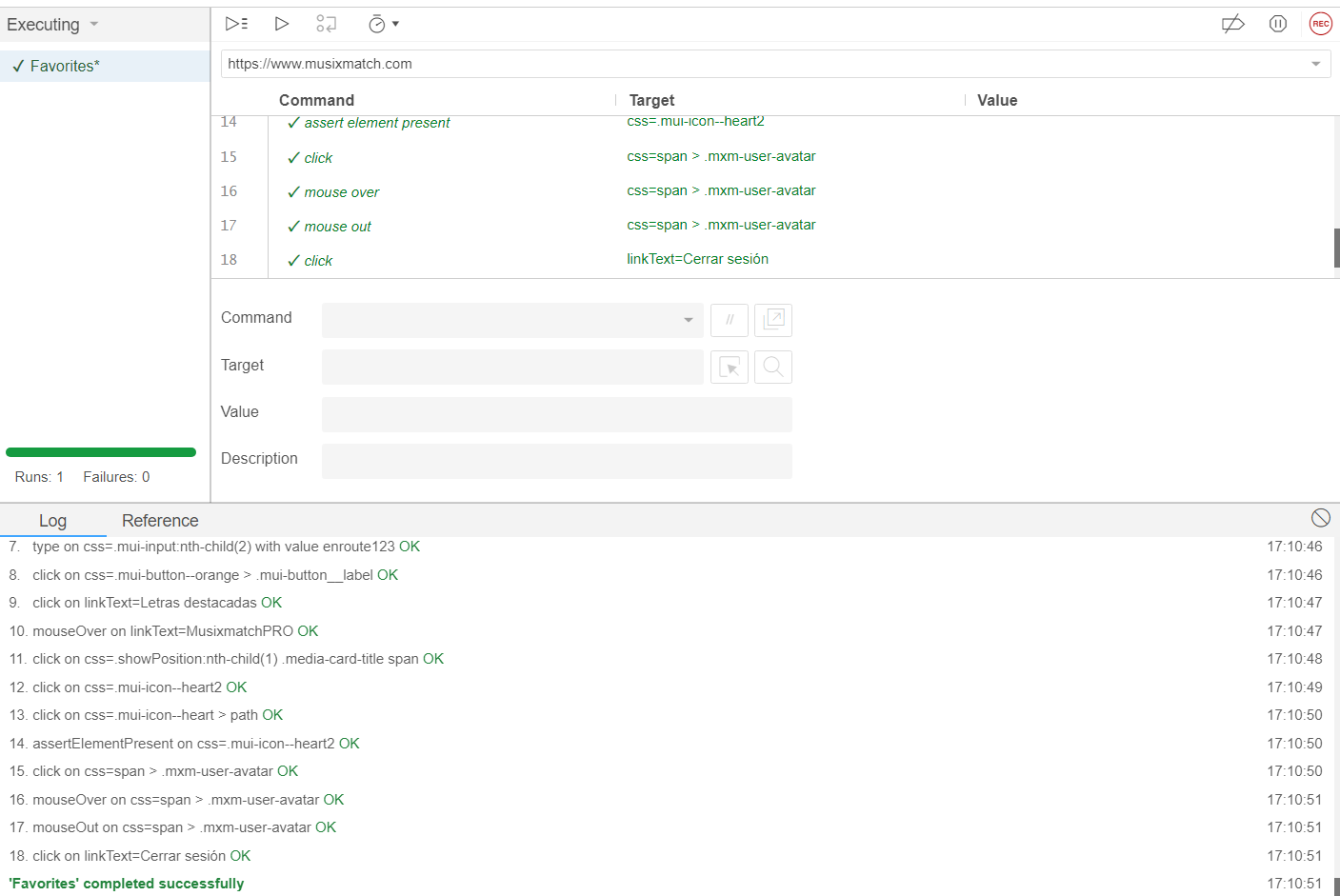
****

**sign in test **

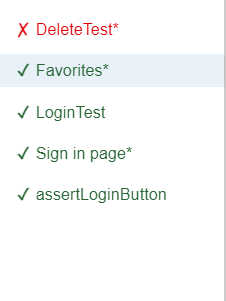
**Log in test**

**Delete test**

**Favorite test**

****

**Test conclusion**



every test worked well except for the delete test since the test wasn't finding the specific button for delete several times.