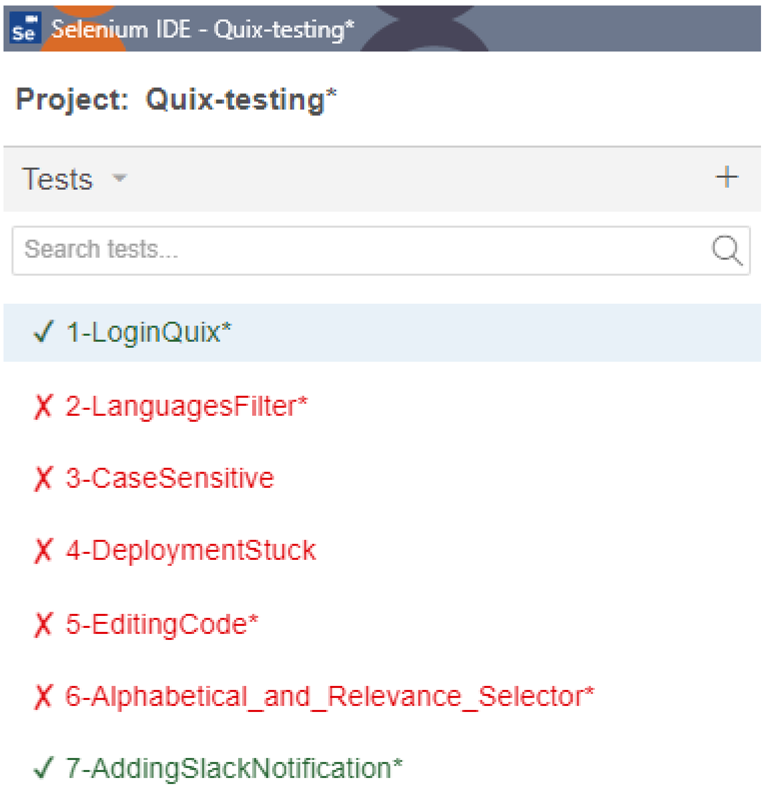
run ./quix_automated_testsuite_v1.0


Selenium IDE - Quix-testing*

Project: Quix-testing*

Tests

Search tests...

✓ 1-LoginQuix*

✗ 2-LanguagesFilter*

✗ 3-CaseSensitive

✗ 4-DeploymentStuck

✗ 5-EditingCode*

✗ 6-Alphabetical_and_Relevance_Selector*

✓ 7-AddingSlackNotification*

1.<login_quix>

Here, the objective is just to simply automate the login for the Quix webpage with the credentials provided.

Of course, this is not part of the scope or doesn't show any issue but it's a good way of starting the test suite.

<scenario>

Access: [redacted]

And use the credentials provided for login

User: [redacted]

Password: [redacted]

<expected result>

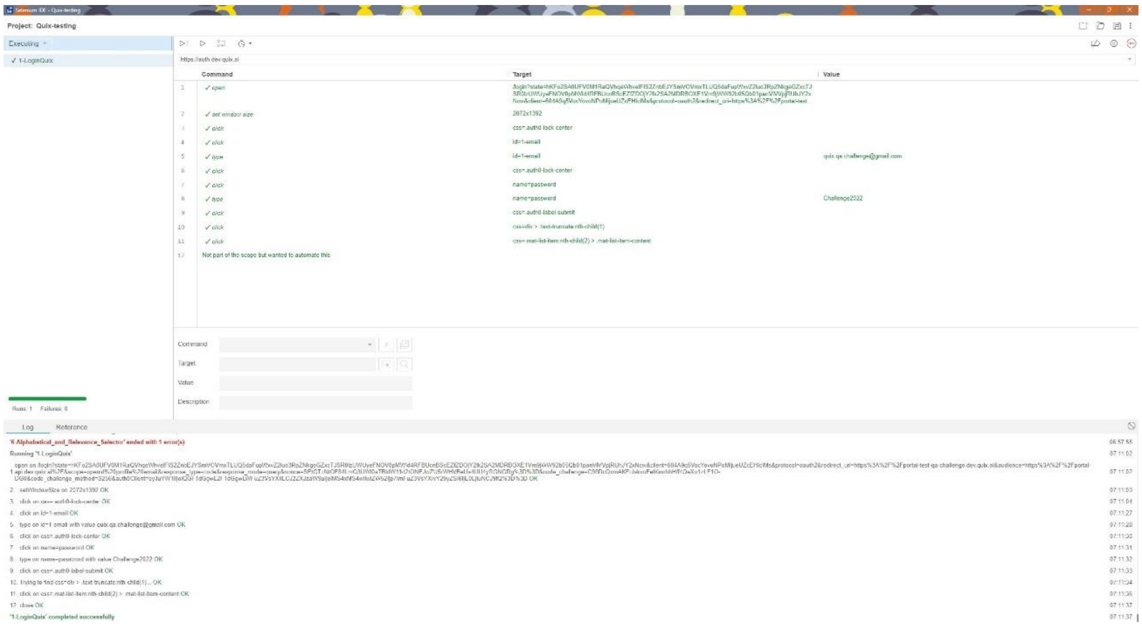
Credentials should work

<actual result>

Login successful

<automation test and screenshot>

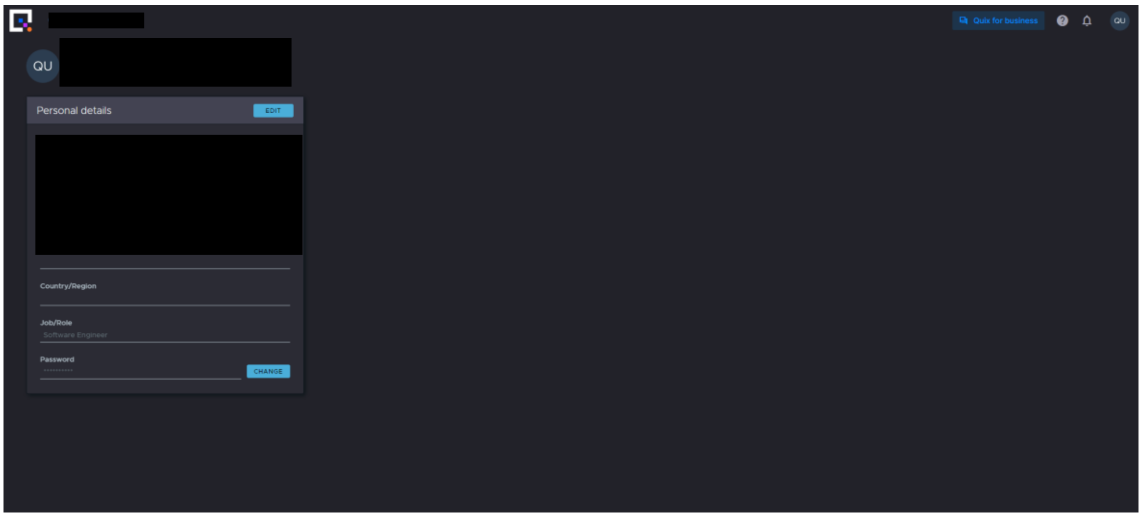
The automated test simply enters the webpage with the provided credential and if the login is successful, the test passes. The fulfilment of the test actions is the assertion per se, as in there's no end assertion to validate the successful login.



	Command	Target	Value
1	✓ open	/login?state=hKFo2SA0UFV0M1RaQVhqeWhvelFIS2ZrbEJYSmVOVmxTLUQ5daFup[illegible]	
2	✓ set window size	2072x1392	
3	✓ click	css=.auth0-lock-center	
4	✓ click	id=1-email	
5	✓ type	id=1-email	quix.qa.challenge@gmail.com
6	✓ click	css=.auth0-lock-center	
7	✓ click	name=password	
8	✓ type	name=password	Challenge2022
9	✓ click	css=.auth0-label-submit	
10	✓ click	css=div > .text-truncate:nth-child(1)	
11	✓ click	css=.mat-list-item:nth-child(2) > .mat-list-item-content	
12	Not part of the scope but wanted to automate this		

```
'6-Alphabetical_and_Relevance_Selector' ended with 1 error(s)                06:57:55
Running '1-LoginQuix'                                                         07:11:02
1  open on /login?state=hKFo2SA0UFV0M1RaQVhqeWhvelFIS2ZrbEJYSmVOVmxTLUQ5daFup[illegible]   07:11:07
2  setWindowSize on 2072x1392 OK                                              07:11:03
3  click on css=.auth0-lock-center OK                                         07:11:04
4  click on id=1-email OK                                                     07:11:27
5  type on id=1-email with value quix.qa.challenge@gmail.com OK               07:11:28
6  click on css=.auth0-lock-center OK                                         07:11:30
7  click on name=password OK                                                  07:11:31
8  type on name=password with value Challenge2022 OK                          07:11:32
9  click on css=.auth0-label-submit OK                                        07:11:33
10 Trying to find css=div > .text-truncate:nth-child(1)... OK                 07:11:34
11 click on css=.mat-list-item:nth-child(2) > .mat-list-item-content OK       07:11:36
12 close OK                                                                   07:11:37
'1-LoginQuix' completed successfully                                          07:11:37
```

<pass screenshot>



[redacted]

QU [redacted]

Personal details

[redacted]

Country/Region

Job/Role
Software Engineer

Password
··········

2.<languages_filter>

In the Library, while trying to filter the programming languages, you’ll see it’s broken.

<scenario>

Should be logged in to Quix webpage.

Go to Library > Select Python, JavaScript, NodeJS, etc…

<expected result>

Filter should work

<actual result>

Filter didn’t work as expected

<workaround>

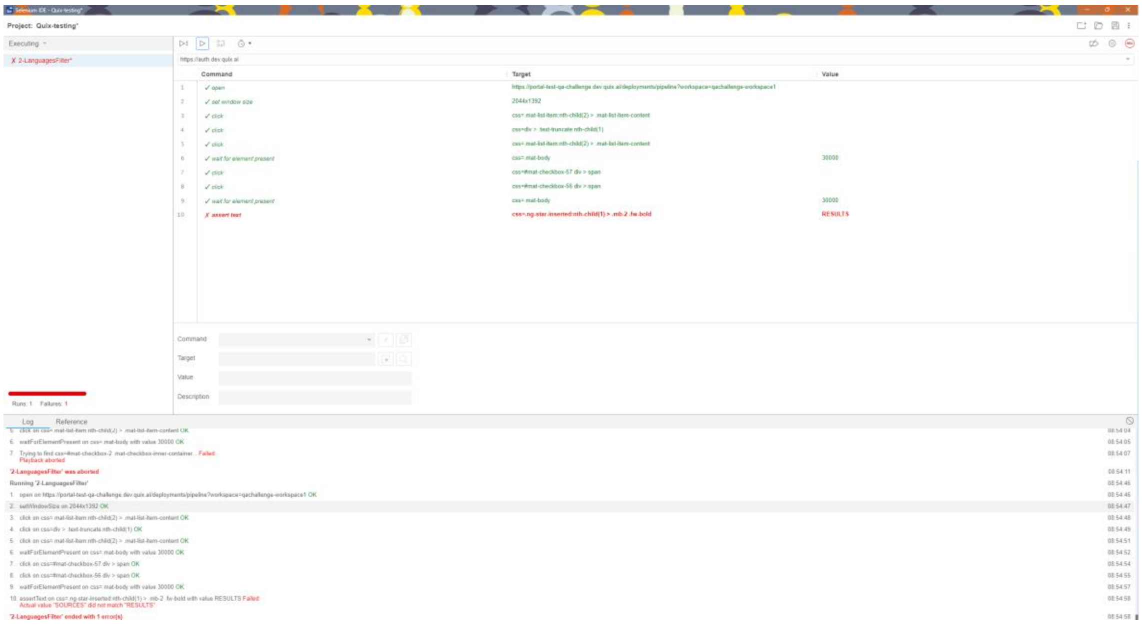
N/A

<provided solution>

Please fix the proper HTML/CSS implementation in the webpage in order to fix this issue.

<automation test and screenshot>

The automated test here enters the webpage already logged in and goes to Library and selects the elements for Shell Script and Python. The assertion is made by *assert text* function, in which the Production Quix changes to RESULTS right after selecting these options. Since it’s not returning this text here, the test fails.



Selenium IDE - Quix-testing*

Project: Quix-testing*

Executing

✗ 2-LanguagesFilter*

https://auth.dev.quix.ai

	Command	Target	Value
1	✓ open	https://portal-test-qa-challenge.dev.quix.ai/deployments/pipeline?workspace=qachallenge-workspace1	
2	✓ set window size	2044x1392	
3	✓ click	css=.mat-list-item:nth-child(2) > .mat-list-item-content	
4	✓ click	css=div > .text-truncate:nth-child(1)	
5	✓ click	css=.mat-list-item:nth-child(2) > .mat-list-item-content	
6	✓ wait for element present	css=.mat-body	30000
7	✓ click	css=#mat-checkbox-57 div > span	
8	✓ click	css=#mat-checkbox-56 div > span	
9	✓ wait for element present	css=.mat-body	30000
10	✗ assert text	css=.ng-star-inserted:nth-child(1) > .mb-2 .fw-bold	RESULTS

Command

Target

Value

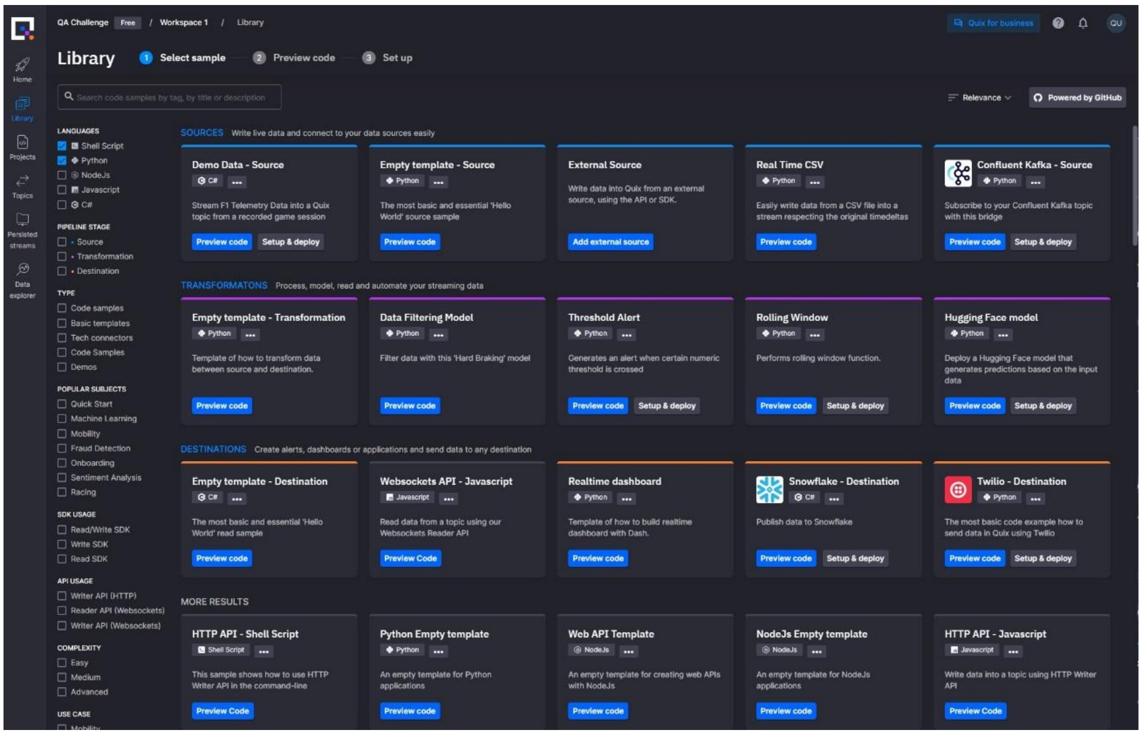
Description

Runs: 1 Failures: 1

Log Reference

```
5.  click on css=.mat-list-item:nth-child(2) > .mat-list-item-content OK                                  08:54:04
6.  waitForElementPresent on css=.mat-body with value 30000 OK                                            08:54:05
7.  Trying to find css=#mat-checkbox-2 .mat-checkbox-inner-container... Failed                            08:54:07
    Playback aborted
'2-LanguagesFilter' was aborted                                                                           08:54:11
Running '2-LanguagesFilter'                                                                               08:54:46
1.  open on https://portal-test-qa-challenge.dev.quix.ai/deployments/pipeline?workspace=qachallenge-workspace1 OK   08:54:46
2.  setWindowSize on 2044x1392 OK                                                                         08:54:47
3.  click on css=.mat-list-item:nth-child(2) > .mat-list-item-content OK                                  08:54:48
4.  click on css=div > .text-truncate:nth-child(1) OK                                                     08:54:49
5.  click on css=.mat-list-item:nth-child(2) > .mat-list-item-content OK                                  08:54:51
6.  waitForElementPresent on css=.mat-body with value 30000 OK                                            08:54:52
7.  click on css=#mat-checkbox-57 div > span OK                                                           08:54:54
8.  click on css=#mat-checkbox-56 div > span OK                                                           08:54:55
9.  waitForElementPresent on css=.mat-body with value 30000 OK                                            08:54:57
10. assertText on css=.ng-star-inserted:nth-child(1) > .mb-2 .fw-bold with value RESULTS Failed           08:54:58
    Actual value "SOURCES" did not match "RESULTS"
'2-LanguagesFilter' ended with 1 error(s)                                                                 08:54:58
```

<fail screenshot>



QA Challenge Free / Workspace 1 / Library

Quix for business

Library ① Select sample ② Preview code ③ Set up

Search code samples by tag, by title or description

Relevance Powered by GitHub

LANGUAGES
- [✓] Shell Script
- [✓] Python
- [ ] NodeJs
- [ ] Javascript
- [ ] C#

PIPELINE STAGE
- [ ] Source
- [ ] Transformation
- [ ] Destination

TYPE
- [ ] Code samples
- [ ] Basic templates
- [ ] Tech connectors
- [ ] Code Samples
- [ ] Demos

POPULAR SUBJECTS
- [ ] Quick Start
- [ ] Machine Learning
- [ ] Mobility
- [ ] Fraud Detection
- [ ] Onboarding
- [ ] Sentiment Analysis
- [ ] Racing

SDK USAGE
- [ ] Read/Write SDK
- [ ] Write SDK
- [ ] Read SDK

API USAGE
- [ ] Writer API (HTTP)
- [ ] Reader API (Websockets)
- [ ] Writer API (Websockets)

COMPLEXITY
- [ ] Easy
- [ ] Medium
- [ ] Advanced

USE CASE
- [ ] Mobility

SOURCES Write live data and connect to your data sources easily

- Demo Data - Source (C#) — Stream F1 Telemetry Data into a Quix topic from a recorded game session — Preview code / Setup & deploy
- Empty template - Source (Python) — The most basic and essential 'Hello World' source sample — Preview code
- External Source — Write data into Quix from an external source, using the API or SDK. — Add external source
- Real Time CSV (Python) — Easily write data from a CSV file into a stream respecting the original timedeltas — Preview code
- Confluent Kafka - Source (Python) — Subscribe to your Confluent Kafka topic with this bridge — Preview code / Setup & deploy

TRANSFORMATONS Process, model, read and automate your streaming data

- Empty template - Transformation (Python) — Template of how to transform data between source and destination. — Preview code
- Data Filtering Model (Python) — Filter data with this 'Hard Braking' model — Preview code
- Threshold Alert (Python) — Generates an alert when certain numeric threshold is crossed — Preview code / Setup & deploy
- Rolling Window (Python) — Performs rolling window function. — Preview code / Setup & deploy
- Hugging Face model (Python) — Deploy a Hugging Face model that generates predictions based on the input data — Preview code / Setup & deploy

DESTINATIONS Create alerts, dashboards or applications and send data to any destination

- Empty template - Destination (C#) — The most basic and essential 'Hello World' read sample — Preview code
- Websockets API - Javascript (Javascript) — Read data from a topic using our Websockets Reader API — Preview Code
- Realtime dashboard (Python) — Template of how to build realtime dashboard with Dash. — Preview code
- Snowflake - Destination (C#) — Publish data to Snowflake — Preview code / Setup & deploy
- Twilio - Destination (Python) — The most basic code example how to send data in Quix using Twilio — Preview code / Setup & deploy

MORE RESULTS

- HTTP API - Shell Script (Shell Script) — This sample shows how to use HTTP Writer API in the command-line — Preview Code
- Python Empty template (Python) — An empty template for Python applications — Preview code
- Web API Template (NodeJs) — An empty template for creating web APIs with NodeJs — Preview code
- NodeJs Empty template (NodeJs) — An empty template for NodeJs applications — Preview code
- HTTP API - Javascript (Javascript) — Write data into a topic using HTTP Writer API — Preview Code

3.<case_sensitive>

While using the search bar in the Library, you’ll see that it’s case sensitive and it shouldn’t be. As in, I want to search for “Transformation” of the Pipelines, and it only shows up if I write it exactly as it’s there. It doesn’t work if I write “transformation”, returns a no result page.

Also, when clicking in the “X” for cleaning the search entry, it doesn’t work.

<scenario>

Should be logged in to Quix webpage.

Go to Library > type “transformation” (not capitalized)

<expected result>

The Transformation pipelines should show up

<actual result>

Search returned no actual results

<workaround>

Type exactly as it’s written, “Transformation”

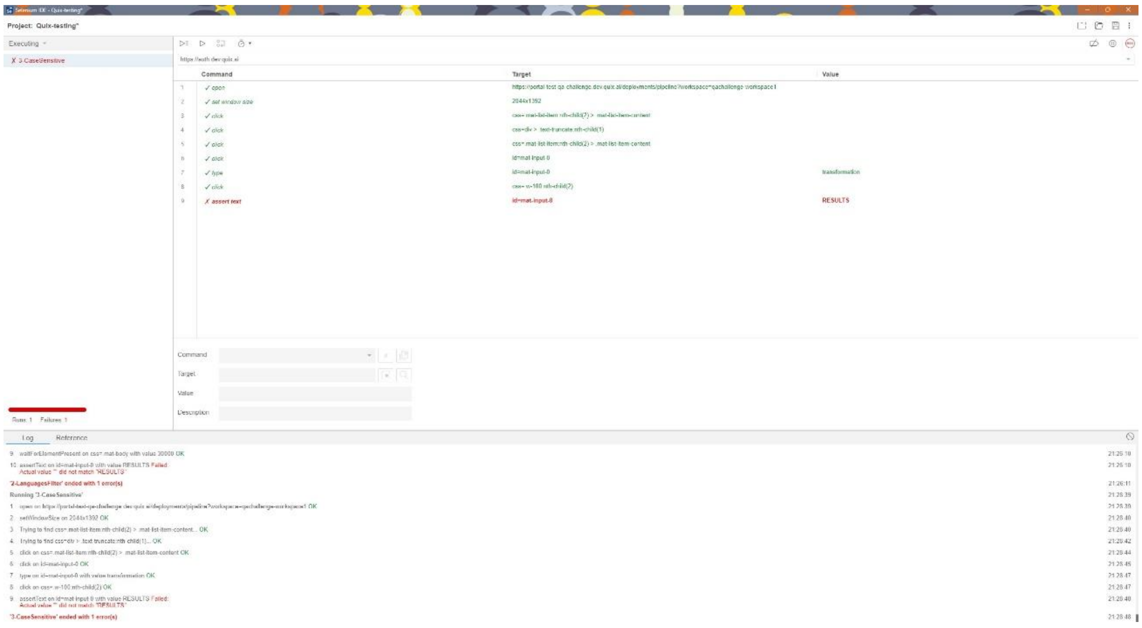
<provided solution>

Apparently, the search bar has some kind of case sensitive filter, while it shouldn’t. The user should write what it wants to write, and his search shouldn’t be affected by this.

Please fix also the “X” for cleaning the search bar after an entry, since it’s not working also.

<automation test and screenshot>

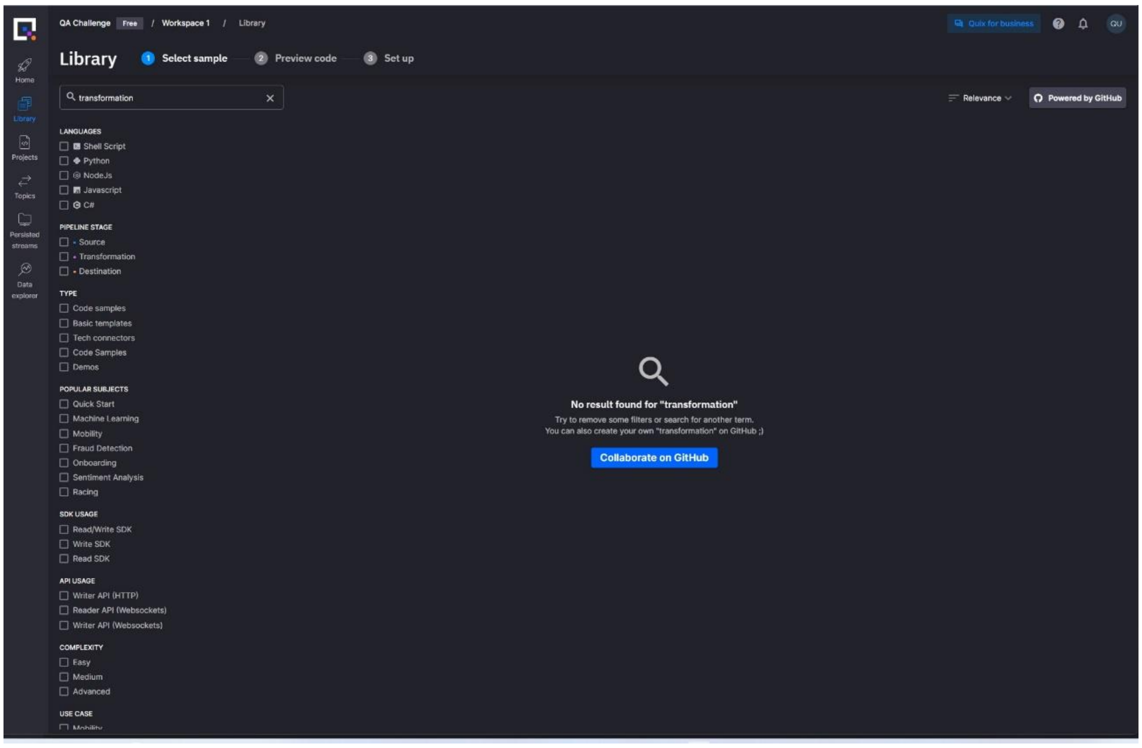
The automated test here enters the webpage already logged in and goes to Library and goes to the search bar on top and writes “transformation”. Since there are no results, the function *assert text* searches for RESULTS (which is the behavior on Quix Prod) and returns a fail.



	Command	Target	Value
1	✓ open	https://portal-test-qa-challenge.dev.quix.ai/deployments/pipeline?workspace=qachallenge-workspace1	
2	✓ set window size	2044x1392	
3	✓ click	css=.mat-list-item:nth-child(2) > .mat-list-item-content	
4	✓ click	css=div > .text-truncate:nth-child(1)	
5	✓ click	css=.mat-list-item:nth-child(2) > .mat-list-item-content	
6	✓ click	id=mat-input-0	
7	✓ type	id=mat-input-0	transformation
8	✓ click	css=.w-100:nth-child(2)	
9	✗ assert text	id=mat-input-0	RESULTS

```
9.  waitForElementPresent on css=.mat-body with value 30000 OK                                          21:26:10
10. assertText on id=mat-input-0 with value RESULTS Failed:                                             21:26:10
    Actual value "" did not match "RESULTS"
'2-LanguagesFilter' ended with 1 error(s)                                                               21:26:11
Running '3-CaseSensitive'                                                                               21:28:39
1.  open on https://portal-test-qa-challenge.dev.quix.ai/deployments/pipeline?workspace=qachallenge-workspace1 OK   21:28:39
2.  setWindowSize on 2044x1392 OK                                                                       21:28:40
3.  Trying to find css=.mat-list-item:nth-child(2) > .mat-list-item-content... OK                       21:28:40
4.  Trying to find css=div > .text-truncate:nth-child(1)... OK                                          21:28:42
5.  click on css=.mat-list-item:nth-child(2) > .mat-list-item-content OK                                21:28:44
6.  click on id=mat-input-0 OK                                                                          21:28:45
7.  type on id=mat-input-0 with value transformation OK                                                 21:28:47
8.  click on css=.w-100:nth-child(2) OK                                                                 21:28:47
9.  assertText on id=mat-input-0 with value RESULTS Failed:                                             21:28:48
    Actual value "" did not match "RESULTS"
'3-CaseSensitive' ended with 1 error(s)                                                                 21:28:48
```

<fail screenshot>



No result found for "transformation"
Try to remove some filters or search for another term.
You can also create your own "transformation" on GitHub ;)

4.<deployment_stuck>

I'm not 100% sure if this should work as I'm expecting, but when deploying any Source, the Quix webpage keeps stuck on “Creating deployment…” and the “Deploy” button keeps loading without ending…

I know that if I click on “Workspace” after the deployment of the source, whatever I’ve deployed is there and working and the Quix Production has the same behavior, so that’s why it’s not really an issue… but this can be marked as a Flow issue, as in the user might expect the page to load to a different one after clicking on the “Deploy” button.

<scenario>

Should be logged in to Quix webpage.

Go to Library > Demo Data – Source > Setup & Deploy > Deploy

<expected result>

The page should refresh and show the user the deployment made

<actual result>

Got stuck after clicking the “Deploy” button and the page didn’t refresh itself

<workaround>

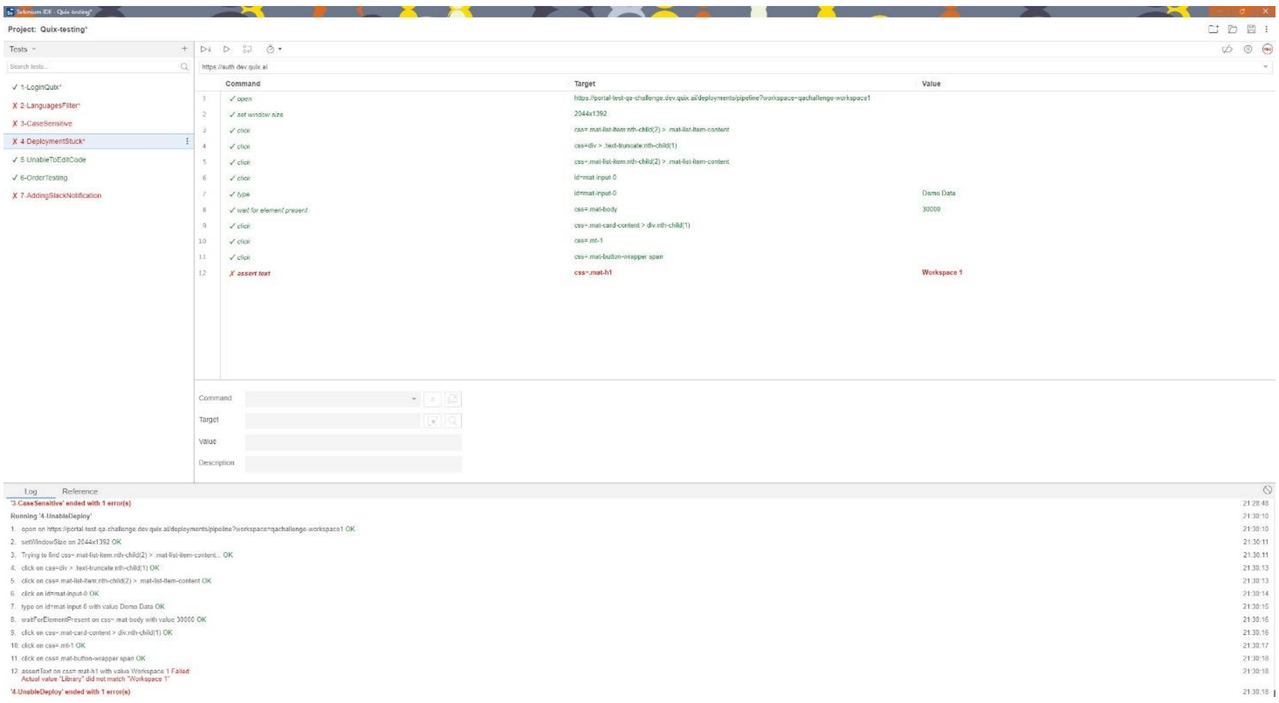
N/A

<provided solution>

The page should be refreshed, showing the user what it deployed.

<automation test and screenshot>

The automated test here enters the webpage already logged in and goes to Library and selects Demo Data – Source, then Setup & Deploy and then Deploy. Since it gets stuck after this option, I asserted the text for Workspace 1, as it’s the page I imagine it should go to. The test searches for the element/text and fails the since it didn’t find it.



Selenium IDE - Quix-testing*

Project: Quix-testing*

Tests

- ✓ 1-LoginQuix*
- ✗ 2-LanguagesFilter*
- ✗ 3-CaseSensitive
- ✗ 4-DeploymentStuck*
- ✓ 5-UnableToEditCode
- ✓ 6-OrderTesting
- ✗ 7-AddingSlackNotification

https://auth.dev.quix.ai

	Command	Target	Value
1	✓ open	https://portal-test-qa-challenge.dev.quix.ai/deployments/pipeline?workspace=qachallenge-workspace1	
2	✓ set window size	2044x1392	
3	✓ click	css=.mat-list-item:nth-child(2) > .mat-list-item-content	
4	✓ click	css=div > .text-truncate:nth-child(1)	
5	✓ click	css=.mat-list-item:nth-child(2) > .mat-list-item-content	
6	✓ click	id=mat-input-0	
7	✓ type	id=mat-input-0	Demo Data
8	✓ wait for element present	css=.mat-body	30000
9	✓ click	css=.mat-card-content > div:nth-child(1)	
10	✓ click	css=.mt-1	
11	✓ click	css=.mat-button-wrapper span	
12	✗ assert text	css=.mat-h1	Workspace 1

Command

Target

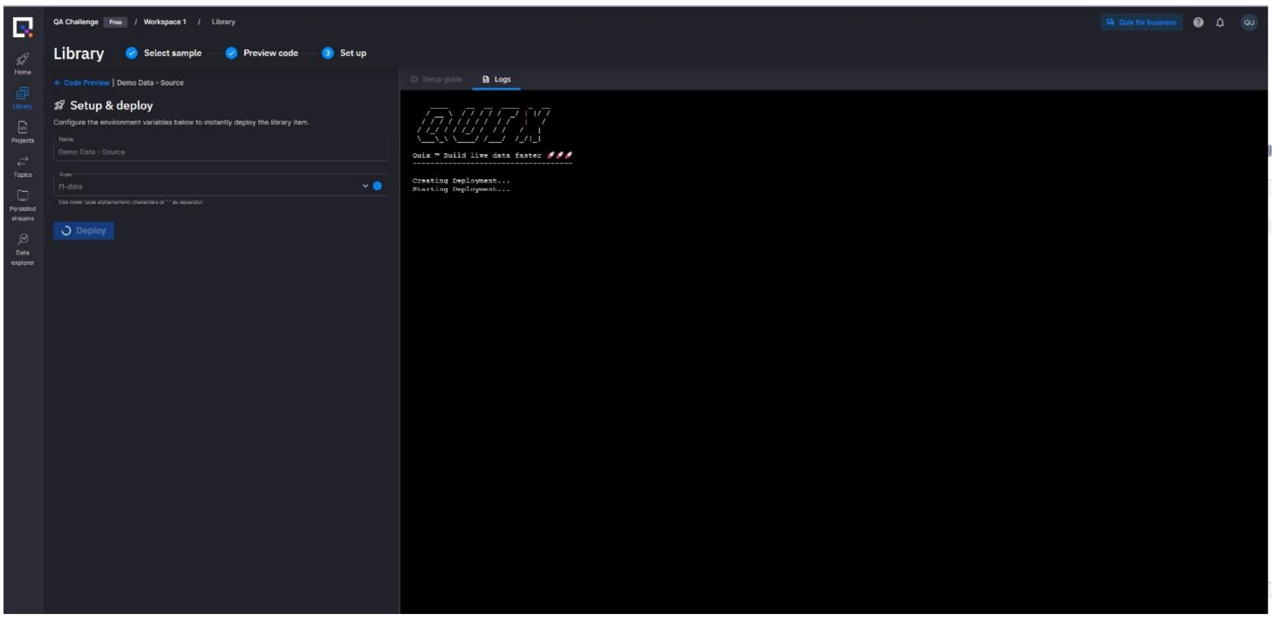
Value

Description

Log Reference

```
'3-CaseSensitive' ended with 1 error(s)                                                          21:28:48
Running '4-UnableDeploy'                                                                          21:30:10
1.  open on https://portal-test-qa-challenge.dev.quix.ai/deployments/pipeline?workspace=qachallenge-workspace1 OK   21:30:10
2.  setWindowSize on 2044x1392 OK                                                                 21:30:11
3.  Trying to find css=.mat-list-item:nth-child(2) > .mat-list-item-content... OK                21:30:11
4.  click on css=div > .text-truncate:nth-child(1) OK                                            21:30:13
5.  click on css=.mat-list-item:nth-child(2) > .mat-list-item-content OK                         21:30:13
6.  click on id=mat-input-0 OK                                                                    21:30:14
7.  type on id=mat-input-0 with value Demo Data OK                                               21:30:15
8.  waitForElementPresent on css=.mat-body with value 30000 OK                                   21:30:16
9.  click on css=.mat-card-content > div:nth-child(1) OK                                         21:30:16
10. click on css=.mt-1 OK                                                                         21:30:17
11. click on css=.mat-button-wrapper span OK                                                      21:30:18
12. assertText on css=.mat-h1 with value Workspace 1 Failed:                                      21:30:18
    Actual value "Library" did not match "Workspace 1"
'4-UnableDeploy' ended with 1 error(s)                                                            21:30:18
```

<fail screenshot>



QA Challenge Free / Workspace 1 / Library

Library Select sample Preview code 3 Set up

← Code Preview | Demo Data - Source

Setup & deploy

Configure the environment variables below to instantly deploy the library item.

Name: Demo Data - Source

Topic: f1-data

Use lower case alphanumeric characters or '-' as separator

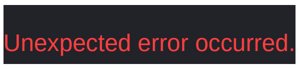
Deploy

Setup guide Logs

```
Quix ™ Build live data faster 🚀🚀🚀
---------------------------------------------

Creating Deployment...
Starting Deployment...
```

5.<editing_code>

This is somewhat similar to the previous test case, but instead of testing the “Setup & Deploy”, the objective here is to test the “Edit code” button. When tried that, and clicked on “Save as project”, I receive an error message: 
Unexpected error occurred.

<scenario>

Should be logged in to Quix webpage.

Go to Library > Demo Data – Source > Edit code > Save as project

<expected result>

The page should refresh and move to “Projects” and allowing the user to edit the code.

<actual result>

Got an error saying “Unexpected error occurred” after clicking the “Save as project” button.

<workaround>

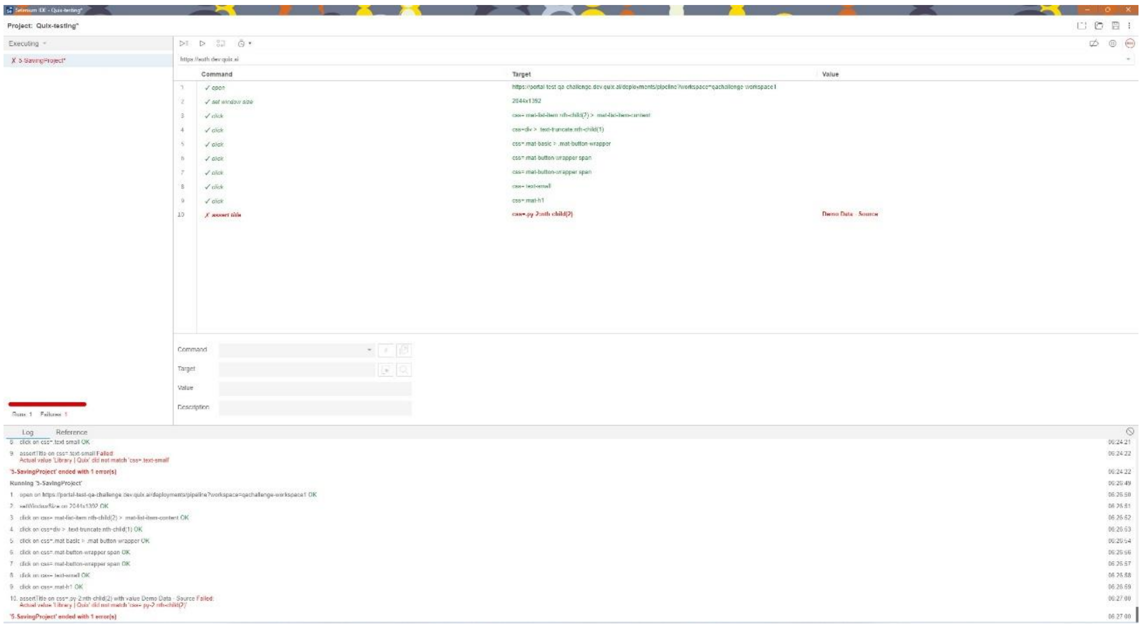
N/A

<provided solution>

The button should work as expected.

<automation test and screenshot>

The automated test here enters the webpage already logged in and goes to Library and selects the Demo Data – Source, then clicks on Edit Code and then Save as Project. Since for Prod Quix, this flow changes the webpage to Project, so I’ve created the assertion on that production element, which here it doesn’t show up – and the test fails, as it should.



Project: Quix-testing*

Executing

✗ 5-SavingProject*

https://auth.dev.quix.ai

	Command	Target	Value
1	✓ open	https://portal-test-qa-challenge.dev.quix.ai/deployments/pipeline?workspace=qachallenge-workspace1	
2	✓ set window size	2044x1392	
3	✓ click	css=.mat-list-item:nth-child(2) > .mat-list-item-content	
4	✓ click	css=div > .text-truncate:nth-child(1)	
5	✓ click	css=.mat-basic > .mat-button-wrapper	
6	✓ click	css=.mat-button-wrapper span	
7	✓ click	css=.mat-button-wrapper span	
8	✓ click	css=.text-small	
9	✓ click	css=.mat-h1	
10	✗ assert title	css=.py-2:nth-child(2)	Demo Data - Source

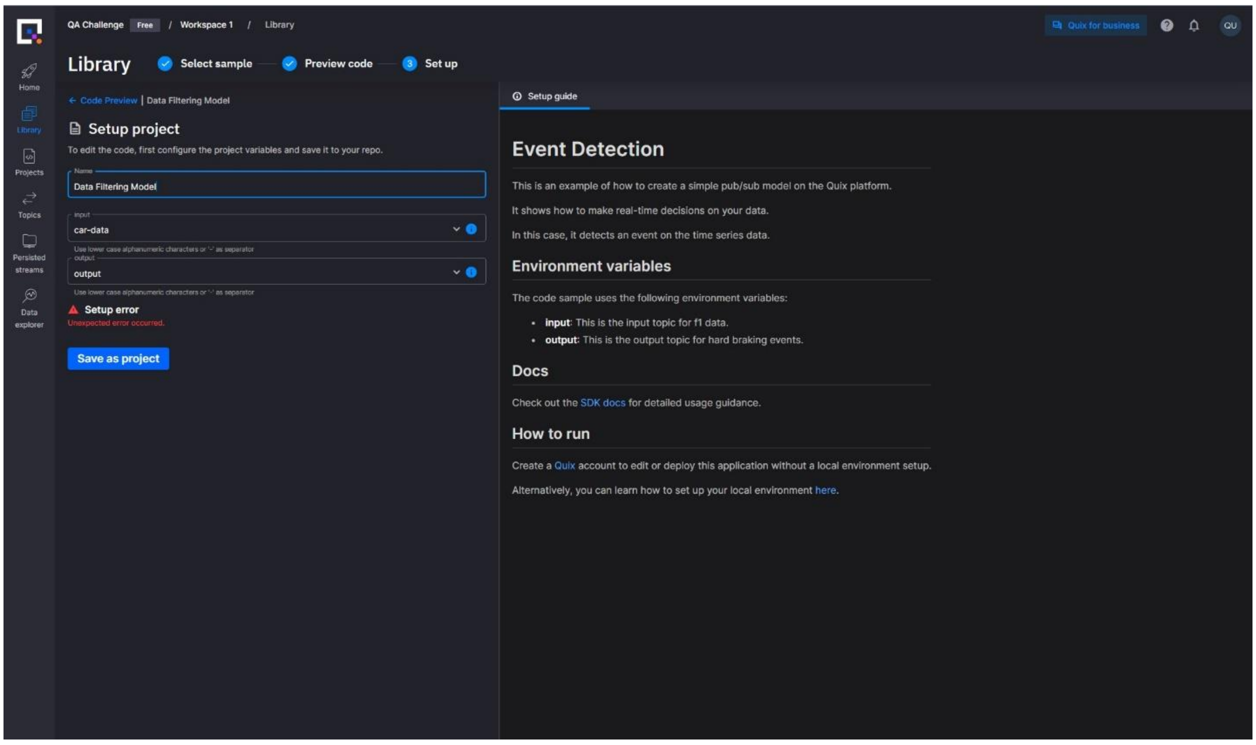
Command
Target
Value
Description

Runs: 1 Failures: 1

Log Reference

```
8.  click on css=.text-small OK                                                                            06:24:21
9.  assertTitle on css=.text-small Failed:                                                                 06:24:22
    Actual value 'Library | Quix' did not match 'css=.text-small'
'5-SavingProject' ended with 1 error(s)                                                                    06:24:22
Running '5-SavingProject'                                                                                  06:26:49
1.  open on https://portal-test-qa-challenge.dev.quix.ai/deployments/pipeline?workspace=qachallenge-workspace1 OK   06:26:50
2.  setWindowSize on 2044x1392 OK                                                                          06:26:51
3.  click on css=.mat-list-item:nth-child(2) > .mat-list-item-content OK                                   06:26:52
4.  click on css=div > .text-truncate:nth-child(1) OK                                                      06:26:53
5.  click on css=.mat-basic > .mat-button-wrapper OK                                                       06:26:54
6.  click on css=.mat-button-wrapper span OK                                                               06:26:56
7.  click on css=.mat-button-wrapper span OK                                                               06:26:57
8.  click on css=.text-small OK                                                                            06:26:58
9.  click on css=.mat-h1 OK                                                                                06:26:59
10. assertTitle on css=.py-2:nth-child(2) with value Demo Data - Source Failed:                            06:27:00
    Actual value 'Library | Quix' did not match 'css=.py-2:nth-child(2)'
'5-SavingProject' ended with 1 error(s)                                                                    06:27:00
```

<fail screenshot>



Library

← Code Preview | Data Filtering Model

Setup project

To edit the code, first configure the project variables and save it to your repo.

Name: Data Filtering Model

input: car-data

Use lower case alphanumeric characters or '-' as separator

output: output

Use lower case alphanumeric characters or '-' as separator

Setup error

Unexpected error occurred.

Event Detection

This is an example of how to create a simple pub/sub model on the Quix platform.

It shows how to make real-time decisions on your data.

In this case, it detects an event on the time series data.

Environment variables

The code sample uses the following environment variables:

- **input**: This is the input topic for f1 data.
- **output**: This is the output topic for hard braking events.

Docs

Check out the SDK docs for detailed usage guidance.

How to run

Create a Quix account to edit or deploy this application without a local environment setup.

Alternatively, you can learn how to set up your local environment here.

6.<alphabetical_selector>

While in Library, if you select the Alphabetical/Relevance dropdown on the top right corner, you'll see that it's not working.

<scenario>

Should be logged in to Quix webpage.

Go to Library > Change the Alphabetical/Relevance dropdown order on the top right corner

Notice any changes

<expected result>

The page should update the order of the components shown in the library, if they are filtered in alphabetical order or relevance

<actual result>

Changing this selector doesn't update anything in the webpage

<workaround>

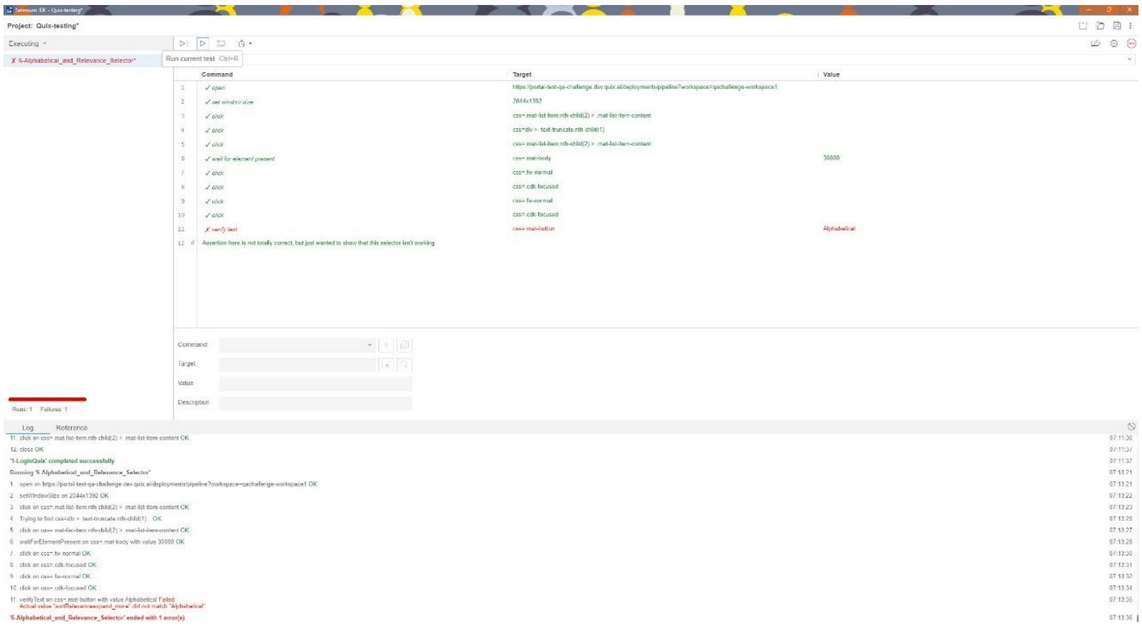
N/A

<provided solution>

The button should work as expected.

<automation test and screenshot>

The automated test here enters the webpage already logged in and goes to Library and selects the dropdown for alphabetical/relevance order that the elements should be displayed. The assertion method here isn't the best one, and it *asserts text* to the previous element in the webpage, but I just wanted to show that this option is failing, so the test could fail as well. Maybe a manual test would do the trick here as well :)


Selenium IDE - Quix-testing*

Project: Quix-testing*

Executing

✗ 6-Alphabetical_and_Relevance_Selector*

Run current test Ctrl+R

	Command	Target	Value
1	✓ open	https://portal-test-qa-challenge.dev.quix.ai/deployments/pipeline?workspace=qachallenge-workspace1	
2	✓ set window size	2044x1392	
3	✓ click	css=.mat-list-item:nth-child(2) > .mat-list-item-content	
4	✓ click	css=div > .text-truncate:nth-child(1)	
5	✓ click	css=.mat-list-item:nth-child(2) > .mat-list-item-content	
6	✓ wait for element present	css=.mat-body	30000
7	✓ click	css=.fw-normal	
8	✓ click	css=.cdk-focused	
9	✓ click	css=.fw-normal	
10	✓ click	css=.cdk-focused	
11	✗ verify text	css=.mat-button	Alphabetical
12	// Assertion here is not totally correct, but just wanted to show that this selector isn't working		

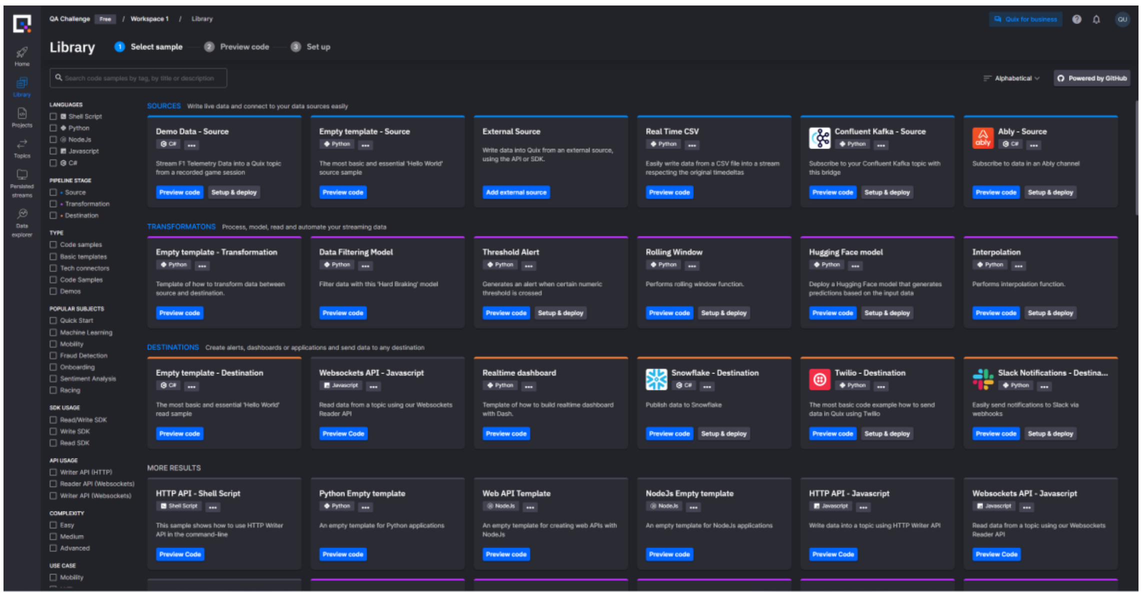
Command
Target
Value
Description

Runs: 1 Failures: 1

Log Reference

```
11. click on css=.mat-list-item:nth-child(2) > .mat-list-item-content OK          07:11:36
12. close OK                                                                     07:11:37
'1-LoginQuix' completed successfully                                             07:11:37
Running '6-Alphabetical_and_Relevance_Selector'                                  07:13:21
1. open on https://portal-test-qa-challenge.dev.quix.ai/deployments/pipeline?workspace=qachallenge-workspace1 OK   07:13:21
2. setWindowSize on 2044x1392 OK                                                 07:13:22
3. click on css=.mat-list-item:nth-child(2) > .mat-list-item-content OK          07:13:23
4. Trying to find css=div > .text-truncate:nth-child(1)... OK                    07:13:25
5. click on css=.mat-list-item:nth-child(2) > .mat-list-item-content OK          07:13:27
6. waitForElementPresent on css=.mat-body with value 30000 OK                    07:13:28
7. click on css=.fw-normal OK                                                    07:13:30
8. click on css=.cdk-focused OK                                                  07:13:31
9. click on css=.fw-normal OK                                                    07:13:32
10. click on css=.cdk-focused OK                                                 07:13:34
11. verifyText on css=.mat-button with value Alphabetical Failed:                07:13:35
    Actual value "sortRelevanceexpand_more" did not match "Alphabetical"
'6-Alphabetical_and_Relevance_Selector' ended with 1 error(s)                    07:13:36
```

<fail screenshot>


QA Challenge Free / Workspace 1 / Library

Library — 1 Select sample — 2 Preview code — 3 Set up

Search code samples by tag, by title or description

Alphabetical — Powered by GitHub — Quix for business

LANGUAGES: Shell Script, Python, NodeJs, Javascript, C#
PIPELINE STAGE: Source, Transformation, Destination
TYPE: Code samples, Basic templates, Tech connectors, Code Samples, Demos
POPULAR SUBJECTS: Quick Start, Machine Learning, Mobility, Fraud Detection, Onboarding, Sentiment Analysis, Racing
SDK USAGE: Read/Write SDK, Write SDK, Read SDK
API USAGE: Writer API (HTTP), Reader API (Websockets), Writer API (Websockets)
COMPLEXITY: Easy, Medium, Advanced
USE CASE: Mobility

SOURCES Write live data and connect to your data sources easily

- Demo Data - Source (C#): Stream F1 Telemetry Data into a Quix topic from a recorded game session. Preview code, Setup & deploy
- Empty template - Source (Python): The most basic and essential 'Hello World' source sample. Preview code
- External Source: Write data into Quix from an external source, using the API or SDK. Add external source
- Real Time CSV (Python): Easily write data from a CSV file into a stream respecting the original timedeltas. Preview code
- Confluent Kafka - Source (Python): Subscribe to your Confluent Kafka topic with this bridge. Preview code, Setup & deploy
- Ably - Source (C#): Subscribe to data in an Ably channel. Preview code, Setup & deploy

TRANSFORMATONS Process, model, read and automate your streaming data

- Empty template - Transformation (Python): Template of how to transform data between source and destination. Preview code
- Data Filtering Model (Python): Filter data with this 'Hard Braking' model. Preview code
- Threshold Alert (Python): Generates an alert when certain numeric threshold is crossed. Preview code, Setup & deploy
- Rolling Window (Python): Performs rolling window function. Preview code, Setup & deploy
- Hugging Face model (Python): Deploy a Hugging Face model that generates predictions based on the input data. Preview code, Setup & deploy
- Interpolation (Python): Performs interpolation function. Preview code, Setup & deploy

DESTINATIONS Create alerts, dashboards or applications and send data to any destination

- Empty template - Destination (C#): The most basic and essential 'Hello World' read sample. Preview code
- Websockets API - Javascript (Javascript): Read data from a topic using our Websockets Reader API. Preview Code
- Realtime dashboard (Python): Template of how to build realtime dashboard with Dash. Preview code
- Snowflake - Destination (C#): Publish data to Snowflake. Preview code, Setup & deploy
- Twilio - Destination (Python): The most basic code example how to send data in Quix using Twilio. Preview code, Setup & deploy
- Slack Notifications - Destina... (Python): Easily send notifications to Slack via webhooks. Preview code, Setup & deploy

MORE RESULTS

- HTTP API - Shell Script (Shell Script): This sample shows how to use HTTP Writer API in the command-line. Preview Code
- Python Empty template (Python): An empty template for Python applications. Preview code
- Web API Template (NodeJs): An empty template for creating web APIs with NodeJs. Preview code
- NodeJs Empty template (NodeJs): An empty template for NodeJs applications. Preview code
- HTTP API - Javascript (Javascript): Write data into a topic using HTTP Writer API. Preview Code
- Websockets API - Javascript (Javascript): Read data from a topic using our Websockets Reader API. Preview Code

7.<adding_slack_notification>

While this test isn’t extremely relevant here, as it works, but just wanted to show an automation I did for adding slack notifications to a project.

<scenario>

Should be logged in to Quix webpage.

User should have a Source deployed and a Transformation already established.

In the Home webpage > Select Add Destination to a pipeline stablished > Slack Notifications > Setup & Deploy > Add webhook_url

<expected result>

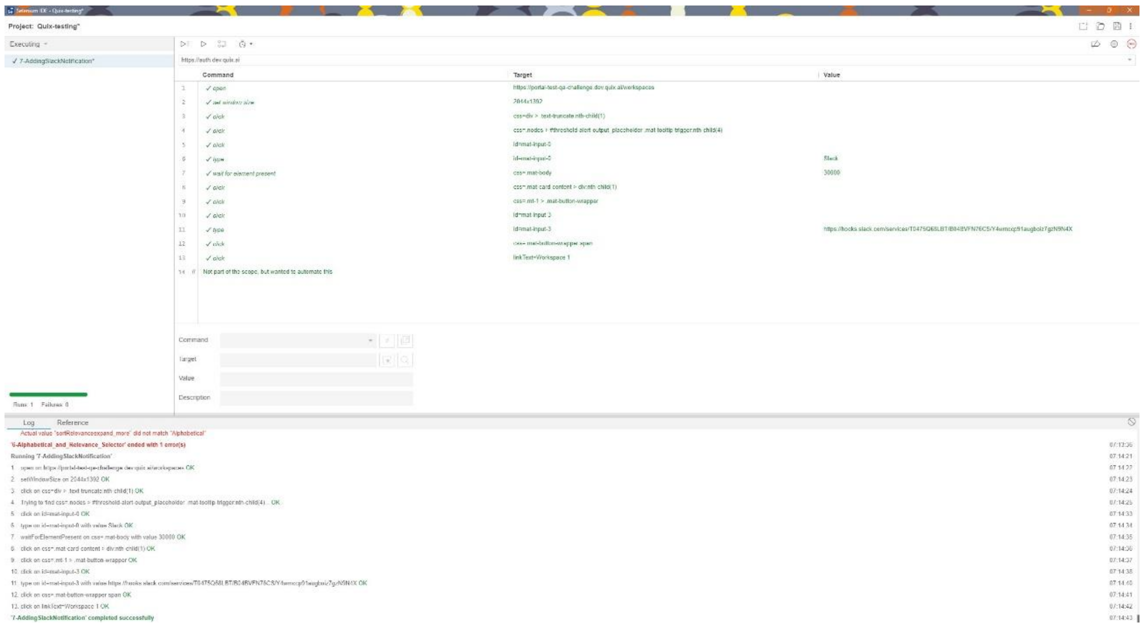
After the integration, your Slack should start receiving the requested data

<actual result>

The integration of Slack notification did work as expected.

<automation test and screenshot>

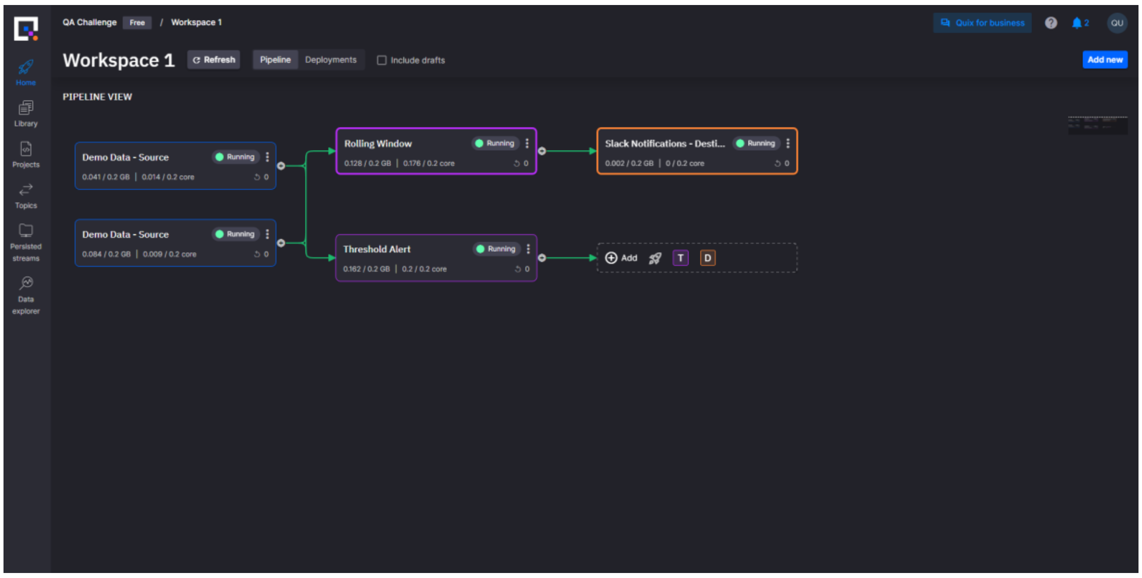
The automated test here enters the webpage already logged and with previous pipeline stablished from Source and Transformation, goes to Home and clicks on the Add Destination for the stablished pipeline. It selects Slack, and adds my own webhook_url to finish the process.



	Command	Target	Value
1	✓ open	https://portal-test-qa-challenge.dev.quix.ai/workspaces	
2	✓ set window size	2044x1392	
3	✓ click	css=div > .text-truncate:nth-child(1)	
4	✓ click	css=.nodes > #threshold-alert-output_placeholder .mat-tooltip-trigger:nth-child(4)	
5	✓ click	id=mat-input-0	
6	✓ type	id=mat-input-0	Slack
7	✓ wait for element present	css=.mat-body	30000
8	✓ click	css=.mat-card-content > div:nth-child(1)	
9	✓ click	css=.mt-1 > .mat-button-wrapper	
10	✓ click	id=mat-input-3	
11	✓ type	id=mat-input-3	https://hooks.slack.com/services/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
12	✓ click	css=.mat-button-wrapper span	
13	✓ click	linkText=Workspace 1	
14	//	Not part of the scope, but wanted to automate this	

```
    Actual value "sortRelevanceexpand_more" did not match "Alphabetical"
'6-Alphabetical_and_Relevance_Selector' ended with 1 error(s)                                  07:13:36
Running '7-AddingSlackNotification'                                                            07:14:21
1. open on https://portal-test-qa-challenge.dev.quix.ai/workspaces OK                          07:14:22
2. setWindowSize on 2044x1392 OK                                                               07:14:23
3. click on css=div > .text-truncate:nth-child(1) OK                                           07:14:24
4. Trying to find css=.nodes > #threshold-alert-output_placeholder .mat-tooltip-trigger:nth-child(4)... OK   07:14:25
5. click on id=mat-input-0 OK                                                                  07:14:33
6. type on id=mat-input-0 with value Slack OK                                                  07:14:34
7. waitForElementPresent on css=.mat-body with value 30000 OK                                  07:14:35
8. click on css=.mat-card-content > div:nth-child(1) OK                                        07:14:36
9. click on css=.mt-1 > .mat-button-wrapper OK                                                 07:14:37
10. click on id=mat-input-3 OK                                                                 07:14:38
11. type on id=mat-input-3 with value https://hooks.slack.com/services/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 OK   07:14:40
12. click on css=.mat-button-wrapper span OK                                                   07:14:41
13. click on linkText=Workspace 1 OK                                                           07:14:42
'7-AddingSlackNotification' completed successfully                                             07:14:43
```

<pass screenshot>



./manual_tests

Not all tests could be automated, so here are some issues that I could find and assert manually.

.<amazon_dynamo_image>

Maybe this isn't extremely relevant (or a very low priority), but while searching for Amazon DynamoDB, the image of the component seems to be broken.

<scenario>

User should be logged in to Quix webpage.

Home webpage > Search for "Amazon" (with the first capital letter) > Look for Amazon DynamoDB

<expected result>

Amazon DynamoDB image and description should appear correctly

<actual result>

The image branding of Amazon DynamoDB is broken

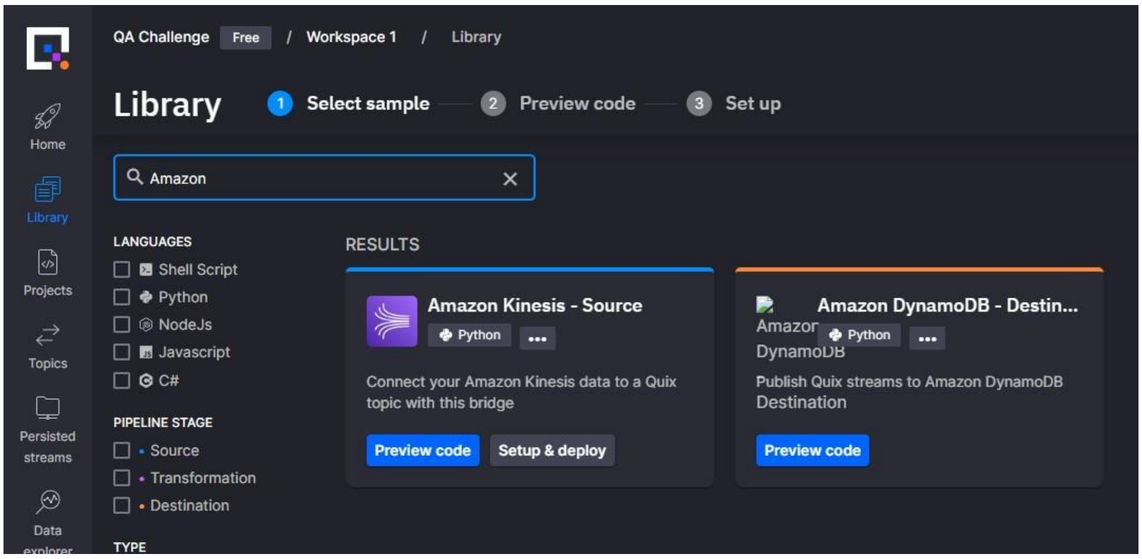
<workaround>

N/A

<provided solution>

Please provide the correct path in the CSS so that the image is properly displayed inside Quix.

<test screenshot>



.<components_mixed>

On Library, when clicking on the Destination of PIPELINE STAGE to the left, you'll see that this brings also some sources and it shouldn't.

<scenario>

User should be logged in to Quix webpage.

Home webpage > select Destination on the left

Notice if there are elements appearing that aren't part of Destination

<expected result>

Only Destination elements should appear in the webpage

<actual result>

Destination and Sources end up appearing

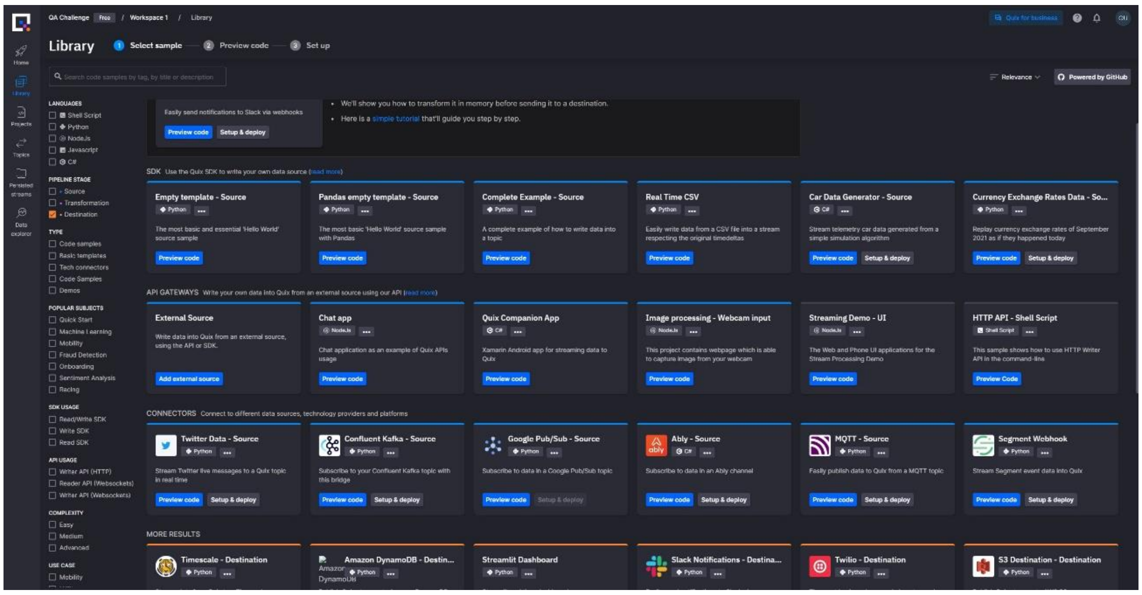
<workaround>

N/A

<provided solution>

Please fix the Destination filter in order to show up only the proper elements marked as such.

<test screenshot>



QA Challenge Free / Workspace 1 / Library

Library ① Select sample — ② Preview code — ③ Set up

Search code samples by tag, by title or description

Relevance · Powered by GitHub

LANGUAGES: ☐ Shell Script ☐ Python ☐ Node.Js ☐ Javascript ☐ C#

PIPELINE STAGE: ☐ Source ☐ Transformation ☑ Destination

TYPE: ☐ Code samples ☐ Basic templates ☐ Tech connectors ☐ Code Samples ☐ Demos

POPULAR SUBJECTS: ☐ Quick Start ☐ Machine Learning ☐ Mobility ☐ Fraud Detection ☐ Onboarding ☐ Sentiment Analysis ☐ Racing

SDK USAGE: ☐ Read/Write SDK ☐ Write SDK ☐ Read SDK

API USAGE: ☐ Writer API (HTTP) ☐ Reader API (Websockets) ☐ Writer API (Websockets)

COMPLEXITY: ☐ Easy ☐ Medium ☐ Advanced

USE CASE: ☐ Mobility

Easily send notifications to Slack via webhooks — Preview code · Setup & deploy

- We'll show you how to transform it in memory before sending it to a destination.
- Here is a simple tutorial that'll guide you step by step.

SDK Use the Quix SDK to write your own data source (read more)

- Empty template - Source (Python): The most basic and essential 'Hello World' source sample
- Pandas empty template - Source (Python): The most basic 'Hello World' source sample with Pandas
- Complete Example - Source (Python): A complete example of how to write data into a topic
- Real Time CSV (Python): Easily write data from a CSV file into a stream respecting the original timedeltas
- Car Data Generator - Source (C#): Stream telemetry car data generated from a simple simulation algorithm
- Currency Exchange Rates Data - So... (Python): Replay currency exchange rates of September 2021 as if they happened today

API GATEWAYS Write your own data into Quix from an external source using our API (read more)

- External Source: Write data into Quix from an external source, using the API or SDK. — Add external source
- Chat app (Node.Js): Chat application as an example of Quix APIs usage
- Quix Companion App (C#): Xamarin Android app for streaming data to Quix
- Image processing - Webcam input (Node.Js): This project contains webpage which is able to capture image from your webcam
- Streaming Demo - UI (Node.Js): The Web and Phone UI applications for the Stream Processing Demo
- HTTP API - Shell Script (Shell Script): This sample shows how to use HTTP Writer API in the command-line

CONNECTORS Connect to different data sources, technology providers and platforms

- Twitter Data - Source (Python): Stream Twitter live messages to a Quix topic in real time
- Confluent Kafka - Source (Python): Subscribe to your Confluent Kafka topic with this bridge
- Google Pub/Sub - Source (Python): Subscribe to data in a Google Pub/Sub topic
- Ably - Source (C#): Subscribe to data in an Ably channel
- MQTT - Source (Python): Easily publish data to Quix from a MQTT topic
- Segment Webhook (Python): Stream Segment event data into Quix

MORE RESULTS

- Timescale - Destination (Python)
- Amazon DynamoDB - Destin... (Python)
- Streamlit Dashboard (Python)
- Slack Notifications - Destina... (Python)
- Twilio - Destination (Python)
- S3 Destination - Destination (Python)

.<cleaning_search_bar>

On Library, type whatever kind of text inside the search bar. Try to click on the “X” of the search bar and notice that it doesn’t do anything.

<scenario>

User should be logged in to Quix webpage.

Home webpage > type any text inside the search bar > Click on the “X” to clean the text

<expected result>

The text should be cleaned after clicking in that button

<actual result>

Button doesn’t do anything

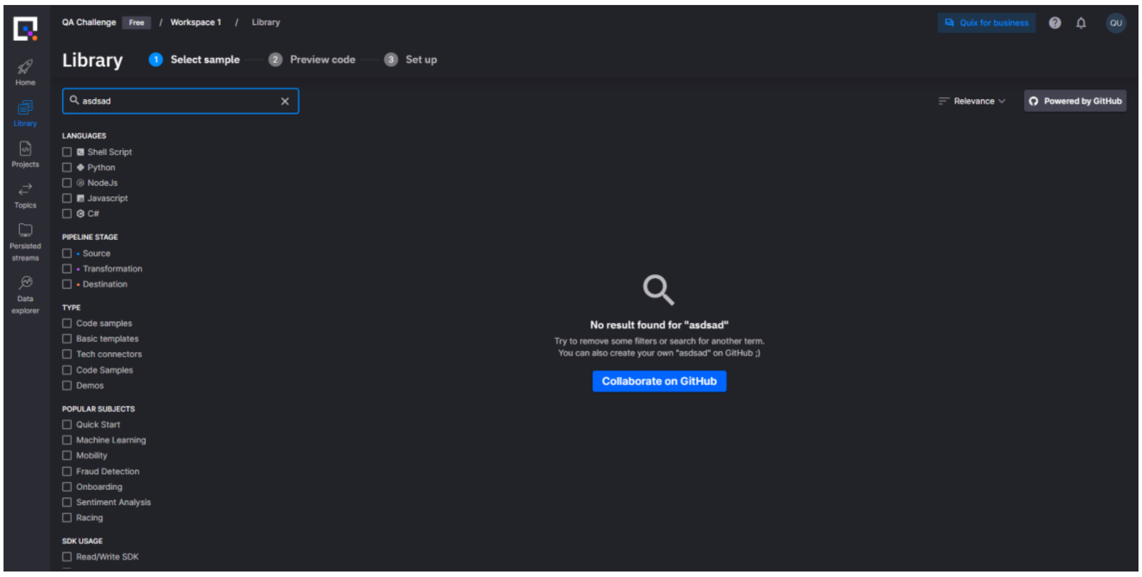
<workaround>

N/A

<provided solution>

Please fix the “X” button on the search bar, so it properly cleans any searches the user do.

<test screenshot>



asdsad

No result found for "asdsad"

Try to remove some filters or search for another term.
You can also create your own "asdsad" on GitHub ;)