Testing Documentation

Testing Environment and Tools

The program was tested on a Linux environment. Shell scripts were used to automate a majority of the testing. Other tools used were sqlite3 for directly checking data from the database.

Limitations

Elements of the API that could not be tested appropriately included proper white box testing of certain conditional flows. Additionally, updating and deleting articles were not tested extensively as they would be used very minimally and can only be accessed by members of the group eliminating any misuse of it.

Testing Procedure

A static database was created with 50 entries. The data was loaded into the database via the scraper. Test scripts were written to change the database accessed by the API to be the static database for the purposes of testing. The test scripts would use curl to perform all the test cases below. The API was run on localhost so as to allow for easy access to the static database. The expected data was manually generated and checked using the database before being placed into the required output files. The test scripts would then perform a diff check on the actual output and expected output. If any discrepancies between outputs were to appear, it would be displayed. Testing of API calls that required authentication were performed manually as it is very difficult to write shell scripts that account for cookies. These were tested by running the command on swagger then checking database and running get articles to see if they were updated or deleted.

All the API endpoints were tested. This included retrieving the latest N amount of articles, searching articles with parameters involving date, location and key terms, and retrieving an article by ID. Additionally, updating and deleting of articles were tested.

Database queries were tested to work in terms of general cases within sqlite3 in the static database. The queries were filled with the required parameters and then executed. The output was checked by hand due to limited number queries to crosscheck.

The scraper was tested manually as well. This involved running the scraper and then comparing the output with the information on the site. The information scraped came from the recent 50 section, so the output from the scraper was compared with that.

Furthermore, each of the articles were opened and its details were compared to the output as well.

Test Cases

General Information

ID refers to the output file name, i.e. 0.actual and 0.expected. These files can be found in the Test Scripts section within the repository. Configuration files can be found for switching the static and normal databases out. Furthermore, running runTests will get all the data and then perform the comparisons between expected and actual output. Any ID with an alphabet was manually tested due to certain complications, i.e. testing for cookies. The AuthCode can be found in API_SourceCode/api/api.js.

Get Articles

ID	Number of Articles Pulled	Reasoning	Pass
0	0	Edge case to check whether no articles can be pulled. Returns an empty list.	✓
1	1	Edge case to check whether one article can be pulled.	✓
2	20	Checks whether default amount is pulled when entering amount.	✓
3	30	General Case	✓
4	34	General Case	✓
5	46	General Case	✓
6	50	Edge case to check whether all entries are returned.	✓
7	60	Edge case to check whether only 50 articles are returned despite number exceeding amount of articles in database.	✓

Get Article by ID

ID	ID of Article	Reasoning	Pass
8	0	Check whether the default article by ID is returned with ID entered. Fails as in this case an article of ID 0 does not exist.	✓
9	1	Edge case to get the second article by ID.	✓
10	5	General case	✓
11	10	General case	✓
12	37	General case	✓
13	49	Edge case to get last article in database.	✓
14	50	Edge case to get article by ID at the end of list of articles.	✓
15	60	Edge case to get article by ID that does not exist.	✓

Search Articles

ID	Parameters	Reasoning	Pass
16	startDate: “” endDate: “” keyTerms: “” Location: “”	Edge case to test no inputs at all. Should return an empty list since start and end date are required.	✓
17	startDate: “2020-04-20T12:00:00” endDate: “” keyTerms: “” Location: “”	Edge case to test a single input. Should return an empty list since the start date is required.	✓
18	startDate: “” endDate: “2020-04-20T12:00:00” keyTerms: “”	Edge case to test a single input. Should return an empty list since the end date is required.	✓

	Location: “”		
19	startDate: “” endDate: “” keyTerms: “COVID-19” Location: “”	Edge case to test a single input. Should return an empty list since start and end date are required.	✓
20	startDate: “” endDate: “” keyTerms: “” Location: “Sydney”	Edge case to test a single input. Should return an empty list since start and end date are required.	✓
21	startDate: “12:00:00 1999-09-09” endDate: “2020-04-20T12:00:00” keyTerms: “COVID-19” Location: “New York”	Edge case incorrect start date. Should return an empty list.	✓
22	startDate: “2018-09-19T09:12:43” endDate: “2021-10-12T10:47:12”	Edge case no key terms and location. Should return a list.	✓
23	startDate: “2018-09-19T09:12:43” endDate: “2021-10-12T10:47:12” Location: “United States”	Edge case no key term only. Should return a list.	✓
24	startDate: “2020-04-20T12:00:00” endDate: “12:00:00 2020-06-10” keyTerms: COVID-19“”	Edge case incorrect end date. Should return an empty list.	✓

	Location: “New York”		
25	startDate: “2020-06-10T12:00:00” endDate: “2020-04-20T12:00:00” keyTerms: “COVID-19” Location: “New York”	Edge case start date greater than end date. Should return an empty list.	✓
26	startDate: “1999-09-09T12:00:00” endDate: “2000-09-09T12:00:00” keyTerms: “COVID-19” Location: “New York”	Edge case date range that does not exist. Should return an empty list.	✓
27	startDate: “2020-01-01T12:00:00” endDate: “2020-04-04T12:00:00” keyTerms: “COVID-19” Location: “Pineapple”	Edge case location that doesn’t exist. Should return an empty list.	✓
28	startDate: “2020-01-01T12:00:00” endDate: “2020-04-04T12:00:00” keyTerms: “Orange Juice” Location: “New York”	Edge case key terms that have no results. Should return a list since it meets other requirements.	✓
29	startDate: “2020-01-01T12:00:00”	Edge case with multiple key terms. Should return a list.	✓

	endDate: “2020-04-04T12:00:00” keyTerms: “COVID-19, other” Location: “New York”		
30	startDate: “2018-09-19T09:12:43” endDate: “2021-10-12T10:47:12” keyTerms: “COVID-19”	Edge case with no location. Should return a list.	✓
31	startDate: “2018-09-19T09:12:43” endDate: “2021-10-12T10:47:12” Location: “Argentina”	General Case	✓
32	startDate: “2018-09-19T09:12:43” endDate: “2021-10-12T10:47:12” keyTerms: “COVID-19” Location: “Italy”	General Case	✓
33	startDate: “2019-01-01T12:00:00” endDate: “2019-01-01T12:01:01” keyTerms: “Influenza” Location: “Hobart”	General Case that produces no results.	✓
34	startDate: “2018-09-19T09:12:4	General Case	✓

	3” endDate: “2021-10-12T10:47:12” keyTerms: “other” Location: “China”		
35	startDate: “2019-12-01T11:01:01” endDate: “2020-04-04T12:01:01” keyTerms: “COVID-19, other” Location: “London”	General Case with multiple key terms.	✓

Update Article

ID	Article ID	Reasoning	Pass
a	1	Edge case without authentication.	✓
b	51	Edge case article ID that doesn’t exist.	✓
c	47	General Case	✓
d	24	General Case	✓

Delete Article

ID	Article ID	Reasoning	Pass
e	1	Edge case without authentication.	✓
f	51	Edge case article ID that doesn’t exist.	✓
g	23	General Case	✓
h	15	General Case	✓