SENG3011 T1 2021

Testing Documentation

Team: Kool Kats

Mentor: Mirette Saleh

Team members:

Lachlan Stewart

z5207906

Pericles Telemachou

z5214738

Stephen Comino

z3470429

Joanne Chen

z5207491

Weiting Han

z5210581

Overview of Testing

In order to ensure our API was working correctly and as expected, there were a few main things to check:

- API Accuracy
  - Our group decided to test our accuracy through manual checking of WHO articles and comparing them to our API output.
- Database accuracy
  - We tested the accuracy of the database through manual checking of WHO articles and what was in the database
- Speed
  - Our group decided to test the speed of our API through the use of the log file/debug output that comes with the API response.

Testing Environment

All tests were done with both a local server and the API server we deployed on AWS. All data and articles used for testing came from the WHO website.

Limitations of testing

Because of the complexity of the data we had to manually check and test our data. We could only test a subset of the data and we assumed correctness with the rest as they were all collected in the same way.

For performance testing we could only test factors that we had direct control over (i.e. time taken for our functions to process requests, time taken to fetch data from database, etc). We couldn't factor in other bottlenecks in the process like time taken to reach the AWS server, application performance (i.e. Postman), etc.

Performance Testing - Manual Checking for API Accuracy

One of the testing processes we used was manual checking for output correctness. This involved going through articles and comparing our results with API output. This would check the accuracy of our API in all endpoints.

We first entered input into our api and checked the resulting output Headlines and article text with the WHO website headlines and article text to see if they are identical.
Because of the large amount of data returned, only a subset of output from the command was tested. This test assumes that all other data will return the same result because the data for other articles are received and processed the same way by the API.

GET /articles

Input (Anthrax United States of America From: 2000-01-10T00:00:00 To:2020-01-10T00:00:00)	API Output From Input	WHO Title:	Comparison of Text	Accuracy
	2001 - Anthrax in the United States (State of Florida)	2001 - Anthrax in the United States (State of Florida)	Exactly the same	100%
	2001 - Anthrax in the United States (State of Florida) - Update	2001 - Anthrax in the United States (State of Florida) - Update	Exactly the same except inline url tag	100%
	2001 - Anthrax in the United States - Update 2	2001 - Anthrax in the United States - Update 2	Exactly the same	100%
	2001 - Anthrax in the United States - Update 3	2001 - Anthrax in the United States - Update 3	Exactly the same	100%
	2001 - Anthrax in the United States - Update 4	2001 - Anthrax in the United States - Update 4	Exactly the same	100%
	2001 - Anthrax in the United States - Update 15	2001 - Anthrax in the United States - Update 15	Exactly the same	100%

The above output is only a subset of the results (too much to put into the table).

Total Accuracy: 100%

Overall, getting the articles and their data (main text, headline, date of publication etc.) through our API has been fairly accurate. It is also accurately mapped to the right locations and diseases. The number of articles found for this case was also accurate (17 articles found).

GET /disease

Input	Manual Checking	API Output	Accuracy
Yellow Fever France From: 2015-01-01T00:00:00 To: 2021-01-01T00:00:00	Cases: 6 Articles: 3	Cases: 19 Articles: 3	Cases: +216% Articles: 100%
Cholera Mexico From: 2013-01-01T00:00:00 To: 2021-01-01T00:00:00	Cases: 184 Articles: 4	Cases: 157 Articles: 1	Cases: -17.2% Articles: -300%
Anthrax From: 2000-01-01T00:00:00 To: 2021-01-01T00:00:00	Cases: 23 Articles: 17	Cases: 45 Articles: 17	Cases: +95.7% Articles: 100%
Dengue Fever From: 2000-01-01T00:00:00 To: 2021-01-01T00:00:00	Cases: 686742 Articles: 37	Cases: 147800 Articles: 21	Cases: -364.6% Articles: -76.2%
Japanese encephalitis From: 2000-01-01T00:00:00 To: 2021-01-01T00:00:00	Cases: 1145 Articles: 1	Cases: 1235 Articles: 1	Cases: +7.9% Articles: 100%

Average Case Accuracy: -12.44%
Average Article Accuracy: -15.24%

Overall, the number of cases and articles have been relatively accurate (with some outliers). However, generally, our API would undercount the articles and overcount cases. Cases have a higher chance of being overcounted as our API finds the number of cases within an article and aggregates it. However, some articles consist of the number of cases 'up to date', causing some errors. Articles on the other hand have a higher chance of being under counted as we match exactly to the disease name and location.

GET /occurrences

Input	Manual Checking	API Output	Accuracy
keyTerms: chikungunya startDate: 2018-01-01T00:00:00 endDate: 2018-12-31T23:59:59	Mombasa, Kenya (1) Sudan (1)	Kenya, Mombasa (1) Sudan (1)	100%
keyTerms: yellow fever startDate: 2017-01-01T00:00:00 endDate: 2019-12-31T23:59:59	Mali (1) Nigeria (5) Bolivarian Republic of Venezuela (1) Brazil (12) Kingdom of the Netherlands (1) Republic of the Congo(1) France (2) Suriname (1)	Brazil (3) France (2) Republic of the Congo (1) Mali (1) Netherlands (1) Nigeria (1) Suriname (1) Venezuela (1)	For “Brazil” and “Nigeria”, there are more than one articles each year, but only count once per year. 6/8 = 75%
keyTerms: Rabies startDate: 2000-01-01T00:00:00 endDate: 2020-12-31T23:59:59	France (1)	France (1)	100%
keyTerms: Hepatitis startDate: 2017-01-01T00:00:00 endDate: 2018-12-31T23:59:59	Namibia (1) Nigeria (1) The Americas and Europe (1) Niger (1) Chad (1)	Chad (1) Namibia (1) Niger (1) The Americas and Europe (1)	4/5 = 80%

Total Accuracy: 88.75%

Overall, there are some inconsistencies between the actual numbers and the number our API returns. However, the accuracy of our endpoint is quite high and would work for rough estimates.

GET /popularDiseases

Input	Manual Checking	API Output	Accuracy
2017-01-01- to 2019-12-31 (France)	Yellow fever: 2017 (1), 2018 (1) Dengue fever: 2018 (1), 2019 (1) Chikungunya: 2017 (1) Rift Valley fever: 2019 (1) Zika virus: 2019 (1)	Dengue fever (2) Yellow fever (2) Chikungunya (1) Rift Valley fever (1)	Zika virus not found in API 6/7 = 86%
2014-01-01 to 2016-12-31 (Spain)	Ebola Virus disease: 2014 (1) Chikungunya: 2015 (2)	Chikungunya (2) Ebola virus disease (1)	3/3 = 100%
2001-01-01 to 2004-12-31 (Nigeria)	2004: Cholera (1), Meningococcal disease (1) 2001: Cholera (2)	Cholera (3) Meningococcal disease (1)	4/4 = 100%
2010-01-01 to 2010-12-31	2010: Influenza (33), Cholera (7), Yellow fever (6), Rift Valley fever (3), Crimean-congo Haemorrhagic fever (1), Meningococcal disease (1), Plague (1)	Influenza (30) Cholera (7) Yellow fever (5) Rift Valley fever (3) Crimean-congo Haemorrhagic fever (1) Meningococcal disease (1) Plague (1)	44/48 = 92%
2002-01-01 to 2002-12-31	2002: Meningococcal (14), Cholera (14), Acute respiratory syndrome (7), Yellow fever (8), Dengue haemorrhagic fever (7), Haemorrhagic fever syndrome (5), Leishmaniasis (4), Plague (2), Acute neurological syndrome (1)	Meningococcal disease (12) Cholera (11) Acute respiratory syndrome (7) Yellow fever (7) Dengue haemorrhagic fever (6) Influenza (6) Haemorrhagic fever syndrome (5) Leishmaniasis (4) Plague (2) Acute neurological syndrome (1)	54/61 = 89%

Total Accuracy: 93.4%

Overall, the accuracy of this endpoint is relatively high.

Manual Checking for Database Accuracy

One of the testing processes we used was manual checking for database and web scraping correctness. This involved going through a few articles and comparing the data, headline and text with the actual WHO article.

Manual Checking Article	API response - Database	Accuracy
Headline: 2001 - Cholera in South Africa Date of Publication: 16 March 2001 Main text: Find at Webpage Location: South Africa Disease: Cholera	Headline: 2001 - Cholera in South Africa Date of Publication: 16 March 2001 Main text: Find at Webpage Location: South Africa Disease: Cholera	100%
Headline: Avian influenza - situation in Egypt - update 39 Date of Publication: *29 December 2010* Main text: Find at Webpage Location: Egypt Disease: Avian Influenza	Headline: Avian influenza - situation in Egypt - update 39 Date of Publication: *29 December 2010* Main text: Find at Webpage Location: Egypt Disease: Avian Influenza	100%
Headline: Cholera in Haiti - update 3 Date of Publication: *17 November 2010* Main text: Find at Webpage Location: Haiti Disease: Cholera	Headline: Cholera in Haiti - update 3 Date of Publication: *17 November 2010* Main text: Find at Webpage Location: Haiti Disease: Cholera	100% - Does not include the attached pdf on the database
Headline: Pandemic (H1N1) 2009 - update 108 Date of Publication: *9 July 2010* Main text: Find at Webpage Location: No Specific location Disease: No Specific location	Headline: Pandemic (H1N1) 2009 - update 108 Date of Publication: *9 July 2010* Main text: Find at Webpage Location: No Specific location Disease: No Specific location	100% - Does not include links and formatted table data. Only TEXT

Total Accuracy: 100%

Overall, all the necessary text is loaded onto the database. However, table data, image data and formatted data do not get loaded onto the database.

Non-Functional Testing - Speed of API response

One of the testing processes we used was manually checking the speed of our API response. This was done by recording the 'service time' property in the log file/debug output that comes with the API response, this measures the time difference between when the request is received at the server and when the response is sent out to the client. By doing this, we were able to measure the average, minimum and maximum time it took for our different request types to be processed at the server.

For each request we tried to maximise the size of the response to simulate a worst case scenario (i.e. broadest date range, not limiting disease names where possible, specifying countries with lots of database entries, etc)

Request Tested	Minimum	Average (of 5)	Maximum
/disease (disease?startDate=1970-01-01&endDate=2021-01-01&location=United States)	22ms (first test)	56ms (first test with 182ms outlier)	182 ms (first test seems like an outlier)
	24ms (second test, 10 requests)	23ms (second test, 10 requests)	24ms (second test, 10 requests)
/articles (articles?startDate=1970-01-01&endDate=2021-01-01&location=United States)	6.2 ms	6.5 ms	7.1 ms
/occurrences (occurrences?keyTerms=hepatitis, cholera, influenza)	5.9 ms	6.0 ms	6.2 ms
/popularDiseases (popularDiseases?startDate=1970-01-01&endDate=2022-01-01&numDiseases=100)	8.9 ms	9.2 ms	9.7 ms

Overall, our responses generally stayed under 30ms. We observed that postman took as little as 300ms and sometimes over 1s to complete a request so there are obviously other bottlenecks outside of our control that extend the time taken to receive a response. A 30ms response will likely be imperceptible to the user and not trigger any timeout errors, therefore we think it's a reasonable amount of time to process an API request at the server.

We observed outliers in the first test for the "disease" endpoint, however, this might be a result of establishing the initial connection or some other rare factor at the server. Subsequent testing produced more consistent results that were more in line with what we were expecting (although they still took at least twice as long as the highest service times from the other endpoints, so there could be further performance optimisations to be made for the "disease" endpoint).

Issues found through testing and improvements made

Through testing, we found errors in our API endpoints and noticed some inconsistencies between our API data and actual data. The main issue we found was that we only added articles to our database if they had a unique article name, which was not true as WHO does contain articles that had duplicate names with differing article content. As we found this close to the deadline, it was too late to scrape and put into our database.

Another issue we found was that not all articles were added to the database. This was due to human error as we had duplicate data and accidentally deleted the database when we tried to remove the duplicates. When we recreated the database from a snapshot the data that we thought we had already added was no longer there. As a result, we did a second round of scraping to get more data for our database.

Another issue we found was that our system for extracting the country name from the data was flawed. To improve this, we went through our data semi-manually in order to correct all the instances of incorrect “Country” values.

Conclusion

Overall, we did testing for database accuracy, API accuracy and speed. Through testing, we were able to find issues and inconsistencies and hence improve our API.