

# API Testing

## Overview

### Environment

Our testing is mainly conducted using Mac OS, except for the testing tools, other client end tests are using both Safari browser and Chrome browser.

### Tools

#### Postman

Postman is a scalable and powerful API testing tool, which can be used to simplify API testing and make the whole process less manually. It can compare the json output with expectation automatically. It allows users to generate a series of API calls and organise them to collections that are convenient to use later. We use postman for both integration tests and load tests.

#### Pytest

Pytest is a python built in framework for unit tests. It provides a rich set of tools for constructing and running tests, which includes aggregation of tests. We mainly use this tool to test our basic functions.

#### LoadView

LoadView is a cloud based real browsers load testing tool. It can simulate common user page interactions and allows us to Define the number of users, user behavior, and duration through custom-built scenarios. We use LoadView to perform load testing and speed testing.

### Process

We tested our API during the entire process of development. Before our scraper actually works, we tested each helper function used during scrapping. And then we individually test the scrapper based on a single page that we already generated the result manually. In the meantime the cloud functions are also separately tested before integration. After the whole API works, we use postman to test both normal case and edge case to ensure we covered all the cases. Finally we did the performance tests and security tests.

## Limitations

The major limitation of our testing is the uncertainty of the results. Since we scripted the pages automatically and periodically, we were unclear of how much results or what results should be returned for each query. It could be more exact if we use a separate database for testing, but due to time limitation, we finally choose to manually check the returned results are correct, and also compare the result with the query of the database to ensure the completeness of the result.

In addition, since the entire CDC website is huge, it is hard to manually check all the page we needed is scripted. We assume that all the reports in CDC have been scripted and stored in the database.

## Correctness Test

### Unit Test

#### Helper Function Test

We use Pytest to test the helper functions used by the scraper, which includes functions to extract location(geonames), date, diseases, syndromes for the paragraph of text.

```
~/3011/SENG3011_2000K/PHASE_1/API_SourceCode(combined*) » pytest
===== test session starts =====
platform darwin -- Python 3.7.3, pytest-6.1.1, py-1.9.0, pluggy-0.13.1
rootdir: /Users/QAQ/3011/SENG3011_2000K/PHASE_1/API_SourceCode
plugins: hypothesis-5.5.4, arraydiff-0.3, remotedata-0.3.2, openfiles-0.4.0, doctestplus-0.5.0,
astropy-header-0.1.2
collected 4 items

helper_test.py .... [100%]

===== 4 passed in 4.12s =====
```

```
import helper

main = "As of November 20, 2020, Fever of unknown Origin 18 people infected"
main2 = "As of November 20, 2020, 18 people infected crimean-congo haemorrhagic fever"

def test_get_date():...
    pass

def test_get_location():
    result1 = helper.get_locations("asdfas asdfasdfa sdf Shanghai sdf")
    result2 = helper.get_locations(main)
    result3 = helper.get_locations("will continue to work with state pu...")
    result4 = helper.get_locations("ted. 6 ill people were children in")

    assert result1[0] == 1796236
    assert result2[0] == 4801859
    print(result3)
    assert result3[0] == 1796236
    assert len(result3) == 1 # test multiple same location only return one
    assert result4[0] == 1814906
    assert result4[1] == 1796236

def test_get_disease():...
    pass

def test_get_syndromes():...
    pass
```

## Scraper Test

To test our scraper, we manually create articles for the certain web pages, scrap that page to verify if the result matches what we want.

## Cloud Function and Database Test

We also tested our cloud function and the database before integrating the entire API. However, both cloud function and database are tested manually in the Google Cloud Platform Console.

For the cloud function, we use the testing function in Cloud Function to manually verify the Output.

METRICS DETAILS SOURCE VARIABLES TRIGGER PERMISSIONS LOGS TESTING

Triggering event 

```
1 {"start_date": "",  
2 "end_date": "",  
3 "location": "china",  
4 "key_terms": "covid"}  
5
```

 TEST THE FUNCTION

Output  Complete

```
$ Require valid start_date and end_date in format yyyy-mm-ddTxx:xx:xx
```



For the database, we use functions to write in or delete data from the schema and check it in the console. And also do some basic query using function and compare the result with the content in the schema.



## Integration Test

For the integration test, we send requests to the API and check the response returned, including status code and response body. It is the black box functional test. Equivalence partitioning, Error guessing, Use Case Analysis and other methods are used for generating these test cases.

### Missing all input parameters

Request Parameters	None
Expected output	The status code returned should be 400 and an error message should also be returned to inform the user
Response time	~2000ms

The screenshot shows the Postman application interface. At the top, there are tabs for '+ New', 'Import', 'Runner', and 'My Workspace'. Below the workspace, a toolbar includes 'Invite', 'No Environment', 'Examples', 'BUILD', 'Send', and 'Save'. A list of recent requests is visible at the top left. The main area shows a 'GET' request to the URL <https://australia-southeast1-seng3011-306108.cloudfunctions.net/cdc-report-api>. The 'Params' tab is selected, displaying four input fields: 'start\_date', 'end\_date', 'location', and 'key\_terms', all of which are currently empty. The 'Body' tab shows a JSON object with 'Key' and 'Value' fields. The response status is 400 Bad Request, with the error message: '1 Require valid start\_date and end\_date in format yyyy-mm-ddTxx:xx:xx'.

All inputs are empty

Input Parameters	<ul style="list-style-type: none"> <li>- <code>start_date: ""</code></li> <li>- <code>end_date: ""</code></li> <li>- <code>location: ""</code></li> <li>- <code>key_terms: ""</code></li> </ul>
Expected output	The status code returned should be 400 and an error message should also be returned to inform the user
Response time	~2000ms

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** Shows 'Collections' (selected), 'History', 'APIs', and a collection named 'API\_TEST' containing 2 requests.
- Request Details:** The 'Missing all parameters' request is selected. It has a GET method and URL: [https://australia-southeast1-seng3011-306108.cloudfunctions.net/cdc-report-api?start\\_date&end\\_date&location&key\\_terms](https://australia-southeast1-seng3011-306108.cloudfunctions.net/cdc-report-api?start_date&end_date&location&key_terms).
- Params Tab:** Shows query parameters: start\_date, end\_date, location, and key\_terms.
- Tests Tab:** Shows a single test: 'Status code is 400' which failed.
- Timeline Tab:** Shows the execution timeline with events like Prepare, Socket Initialization, DNS Lookup, TCP Handshake, SSL Handshake, Transfer Start, Download, Process, and Total time of 2.11s.
- Bottom Navigation:** Includes 'Find and Replace', 'Console', 'Bootcamp', 'Build', 'Browse', and other icons.

## Empty start\_date and end\_date

Input Parameters	<ul style="list-style-type: none"> <li>- <b>start_date:</b></li> <li>- <b>end_date:</b></li> <li>- <b>location:</b> "Sydney"</li> <li>- <b>key_terms:</b> "covid-19"</li> </ul>
Expected Output	The status code returned should be 400 and an error message should also be returned to inform the user
Response Time	~2000ms

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** Collections (selected), APIs, History, + New Collection, API\_TEST (2 requests), GET Missing all parameters, GET Missing only start\_date & en..., test (9 requests).
- Request Details:** GET https://australia-southeast1-seng3011-306108.cloudfunctions.net/cdc-report-api?start\_date&end\_date&location=Sdney&key\_terms=covid-19
- Params:**

KEY	VALUE	DESCRIPTION
start_date		
end_date		
location	Sdney	
key_terms	covid-19	
Key	Value	Description
- Test Results:** Status: 400 Bad Request, Time: 1988 ms, Size: 489 B. One test step is listed: PASS Status code is 400.
- Timeline:** Shows the execution steps and their times:
 

EVENT	TIME
Prepare	1.11 ms
Socket Initialization	1.52 ms
DNS Lookup	Cache
TCP Handshake	Cache
SSL Handshake	Cache
Transfer Start	1984.68 ms
Download	1.23 ms
Process	0.08 ms
Total	1988.6 ms

## Invalid date input

Input Parameters	<ul style="list-style-type: none"> <li>- start_date: "123456"</li> <li>- end_date: "zxcv-zx-tgT12:12:12"</li> <li>- location:</li> <li>- key_terms:</li> </ul>
Expected Output	The status code should be 400 and an error message should be returned to inform the user
Response Time	~2000ms

The screenshot shows the Postman application interface. At the top, there are tabs for '+ New', 'Import', 'Runner', and 'My Workspace'. The 'My Workspace' tab is selected. In the center, there are three error status indicators: 'GET Missing all parameters' (green), 'GET Missing only start\_date & end...' (orange), and 'GET Invalid date format' (green). Below these, a search bar shows 'Invalid date format'. The main workspace displays a 'GET' request to 'https://australia-southeast1-seng3011-306108.cloudfunctions.net/cdc-report-api-geo?start\_date=123456&end\_date=zxcv-zx-tgT12:12:12&location&key\_terms'. The 'Params' tab is active, showing query parameters: start\_date (123456), end\_date (zxcv-zx-tgT12:12:12), location, and key\_terms. The 'Body' tab shows a JSON payload: { "start\_date": "123456", "end\_date": "zxcv-zx-tgT12:12:12", "location": "", "key\_terms": "" }. The 'Test Results' section shows one failure: 'Require valid start\_date and end\_date in format yyyy-mm-ddTxx:xx:xx'. The 'Timeline' section on the right shows the request took 2.27 seconds, with most time spent in Transfer Start (2.27 ms) and DNS Lookup (2.52 ms). The bottom navigation includes 'Find and Replace', 'Console', 'Bootcamp', 'Build', 'Browse', and other icons.

end\_date is before start\_end

Input Parameters	<ul style="list-style-type: none"> <li>- start_date: "2015-08-06T12:34:45"</li> <li>- end_date: "2010-03-25Txx:xx:xx"</li> <li>- location:</li> <li>- key-terms:</li> </ul>
Expected output	The status code returned should be 400 and an error message should also be returned to inform the user
Response time	~2000ms

The screenshot shows the Postman application interface. At the top, there are several status indicators: 'GET Missing all parameters' (red), 'GET Missing only start\_date & end...' (orange), 'GET Invalid date format' (green), and 'GET start\_date > end\_date' (red). Below these, the URL is set to `https://australia-southeast1-seng3011-306108.cloudfunctions.net/cdc-report-api?start_date=2015-08-06T12:34:45&end_date=2010-03-25Txx:xxxx&location&key_terms`. The 'Send' button is highlighted in blue. The 'Params' tab is selected, showing the following query parameters:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> start_date	2015-08-06T12:34:45	
<input checked="" type="checkbox"/> end_date	2010-03-25Txx:xxxx	
<input checked="" type="checkbox"/> location		
<input checked="" type="checkbox"/> key_terms		

The 'Body' tab shows the response body: '1 end\_date must be after start\_date'. On the right, a timeline chart details the request's execution: Event (Prepare, 2.38 ms), Socket Initialization (1.67 ms), DNS Lookup (Cache, 0 ms), TCP Handshake (Cache, 0 ms), SSL Handshake (Cache, 0 ms), Transfer Start (2.25 s), Download (2.22 ms), Process (0.17 ms), and Total (2.25 s).

## Multiple key terms

Input Parameters	<ul style="list-style-type: none"> <li>- <code>start_date: "2019-11-01Txx:xx:xx"</code></li> <li>- <code>end_date: "2020-01-01Txx:xx:xx"</code></li> <li>- <code>location: ""</code></li> <li>- <code>key_terms: "listeria,ebola,salmonella"</code></li> </ul>
Expected Response	API should return a list of reports which have publication date in between <code>start_date</code> and <code>end_date</code> and contains AT LEAST one key term listed in <code>key_terms</code>
Response Time	~3500ms

The screenshot shows a Postman workspace with several failed requests at the top (Missing all parameters, Missing start\_date, Invalid date format, Search for multiple key terms). Below is a successful GET request to a CDC report API. The request URL includes parameters for start\_date, end\_date, location, and key\_terms. The response is a 200 OK with a JSON payload containing an array of articles, each with a headline, date of publication, and a detailed main text about a Listeria infection outbreak.

## No articles returned

Input Parameters	<ul style="list-style-type: none"> <li>- <code>start_date: "2019-11-01Txx:xx:xx"</code></li> <li>- <code>end_date: "2020-01-01Txx:xx:xx"</code></li> <li>- <code>location: "sydney"</code></li> <li>- <code>key_terms: "asdfsadf"</code></li> </ul>
Expected Response	The status code returned should be 200 and an empty list should be returned
Response Time	

## Case Insensitive

### Lower Case

Input Parameters	<ul style="list-style-type: none"> <li>- <code>start_date: "2019-11-01Txx:xx:xx"</code></li> <li>- <code>end_date: "2020-01-01Txx:xx:xx"</code></li> <li>- <code>location: "canada"</code></li> </ul>
------------------	---



The screenshot shows the Postman interface with the following details:

- Header Bar:** + New, Import, Runner, My Workspace, Invite, No Environment, Upgrade.
- Request URL:** https://australia-southeast1-seng3011-306108.cloudfunctions.net/cdc-report-api?start\_date=2019-11-01Txx:xx:xx&end\_date=2020-01-01Txx:xx:xx&location=canada&key\_terms=...
- Params Tab:** start\_date: 2019-11-01Txx:xx:xx, end\_date: 2020-01-01Txx:xx:xx, location: canada, key\_terms: (empty).
- Body Tab:** Status: 200 OK, Time: 3.60 s, Size: 6.32 KB. The response body is a JSON object containing outbreak details for Canada.

```

18     "geonames_id": 6251999
19   }
20 }
21 ]
22 ],
23   "url": "https://www.cdc.gov/listeria/outbreaks/monocytogenes-08-19/index.html"
24 }
25 },
26   "log": {
27     "access_time": "2021-03-26 05:25:57",
28     "data_source": "Centers for Disease Control and Prevention",
29     "execution_time": 2.7904436588287354,
30     "report_found": 1,
31     "team_name": "SENG3011 2000K"
32   }
33
  
```

## Capital

Input Parameters	<ul style="list-style-type: none"> <li>- start_date: "2019-11-01Txx:xx:xx"</li> <li>- end_date: "2020-01-01Txx:xx:xx"</li> <li>- location: "CANADA"</li> <li>- key_terms: ""</li> </ul>
Expected Response	The status code should be 200 and the results should be same as query "sydney"
Response Time	~3000ms

The screenshot shows the Postman interface with the following details:

- Header Bar:** My Workspace, Invite, Upgrade.
- Request List:** Untitled Request, GET https://australia-southeast1-seng3011-306108.cloudfunctions.net/cdc-report-api?start\_date=2019-11-01Txxxxxx&end\_date=2020-01-01Txxxxxx&location=CANADA&key\_terms=... (highlighted), GET https://australia..., GET start\_date > end..., +, etc.
- Environment:** No Environment
- Request Details:** Method: GET, URL: https://australia-southeast1-seng3011-306108.cloudfunctions.net/cdc-report-api?start\_date=2019-11-01Txxxxxx&end\_date=2020-01-01Txxxxxx&location=CANADA&key\_terms=...  
Params: start\_date: 2019-11-01Txxxxxx, end\_date: 2020-01-01Txxxxxx, location: CANADA, key\_terms: (empty)  
Query Params: Key: Value: Description
- Body Response:**

```
1 {  
2   "articles": [  
3     {  
4       "date_of_publication": "2019-11-07 xx:xx:xx",  
5       "headline": "Outbreak of Listeria Infection",  
6       "main_text": "Outbreak of Listeria InfectionsFinal Update \n\nPosted November 7, 2019 at 1:00 PM ET\n\nThis outbreak appears to be over. CDC, several states, and federal partners investigated a multistate outbreak of Listeria monocytogenes infections. A single, common source of infections was not identified. Learn more about how you can take steps to prevent Listeria infection.\n\nFinal Outbreak Information\n\nAt a Glance\nReported Cases: 24\nStates: 13\nHospitalizations: 22\nDeaths: 2\n\n\n\n\n\n\n\n24 people infected with the outbreak strain of Listeria monocytogenes were reported from 13 states. No additional ill people were reported after the initial investigation notice on August 23, 2019.\n\nOf 23 ill people with information available, 22 hospitalizations were reported.\n\nTwo deaths were reported.\n\nA specific food item, grocery store, food processor, or restaurant chain was not identified as the source of infections in the United States.\n\nThe outbreak appears to be over.\n\nAdvice to Consumers\n\nListeria is most likely to sicken pregnant women and their newborns, adults aged 65 or older, and people with weakened immune systems. People who have underlying medical conditions or take medicines (like steroids or chemotherapy) that lower the body's ability to fight germs and sickness have weakened immune systems.\n\nSome foods are more likely to contain Listeria. People at higher risk for infection should:\n\nHeat lunch meats, cold cuts, or other deli meats to an internal temperature of 165°F or until steaming hot just before serving.\n\nAvoid eating cold smoked fish, unless it is canned or shelf-stable or it is in a cooked dish, such as a casserole.\n\nA food is shelf-stable if it can be safely stored at room temperature.\n\nChoose pasteurized milk and milk products, including soft cheese, ice cream, and yogurt.\n\nSymptoms of Listeria Infection\n\nListeriosis can cause different symptoms, depending on the person and the part of the body affected.\n\nPregnant women typically experience only fever and other flu-like symptoms, such as fatigue and muscle aches. However, infections
```
- Bottom Bar:** Find and Replace, Console, Status: 200 OK, Time: 3.16 s, Size: 6.32 KB, Save Response.

The screenshot shows the Postman interface with the following details:

- Header Bar:** + New, Import, Runner, My Workspace (Untitled Request), Invite, Upgrade.
- Request List:** GET Missing all para..., GET Missing only st..., GET Invalid date for..., GET Search for mult..., GET Return no results, GET https://australia..., GET start\_date > end..., and others.
- Environment:** No Environment.
- Request Details:**
  - Method:** GET
  - URL:** https://australia-southeast1-seng3011-306108.cloudfunctions.net/cdc-report-api?start\_date=2019-11-01Txx:xx:xx&end\_date=2020-01-01Txx:xx:xx&location=CANADA&key\_terms=.
  - Params:** (selected) Authorization, Headers (6), Body, Pre-request Script, Tests, Settings.
  - Query Params:**

KEY	VALUE	DESCRIPTION
start_date	2019-11-01Txx:xx:xx	
end_date	2020-01-01Txx:xx:xx	
location	CANADA	
key_terms		
  - Body:** Cookies, Headers (7), Test Results.
  - Response Preview:**

```

16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
        },
        {
            "geonames_id": 6251999
        }
    ],
    "url": "https://www.cdc.gov/listeria/outbreaks/monocytogenes-08-19/index.html"
},
"log": {
    "access_time": "2021-03-26 05:27:31",
    "data_source": "Centers for Disease Control and Prevention",
    "execution_time": 2.5121214389801025,
    "report_found": 1,
    "team_name": "SENG3011 2000K"
}
}

```
  - Status:** Status: 200 OK Time: 3.16 s Size: 6.32 KB Save Response.
- Bottom Navigation:** Find and Replace, Console, Bootcamp, Build, Browse.

## Hierarchy

Input Parameters	<ul style="list-style-type: none"> <li>- start_date: "2018-11-01Txx:xx:xx"</li> <li>- end_date: "2020-01-01Txx:xx:xx"</li> <li>- location: "US"</li> <li>- key_terms: ""</li> </ul>
Expected Response	The status code should be 200 and the results should contain the articles whose reports have location in the United States, e.g. "New York".
Response Time	~4000ms

The screenshot shows the Postman application interface. At the top, there are tabs for '+ New', 'Import', 'Runner', and 'My Workspace'. The 'My Workspace' tab is selected. On the right, there are icons for 'Invite', 'Upgrade', and environment management. Below the header, a list of recent requests is shown, including 'GET Missing al...', 'GET Missing ...', 'GET Invalid da...', 'GET Search f...', 'GET Return no...', 'GET Case Ins...', 'GET start\_date...', 'GET Case Ins...', 'GET Location ...', and 'GET Case Ins...'. A 'No Environment' button is present. To the right, there are buttons for 'Examples', 'BUILD', 'Send', 'Save', and 'Code'. The main area shows a request configuration for a GET request to [https://australia-southeast1-seng3011-306108.cloudfunctions.net/cdc-report-api?start\\_date=2018-11-01Txxxxxx&end\\_date=2020-01-01Txxxxxx&location=US&key\\_terms=mers-cov](https://australia-southeast1-seng3011-306108.cloudfunctions.net/cdc-report-api?start_date=2018-11-01Txxxxxx&end_date=2020-01-01Txxxxxx&location=US&key_terms=mers-cov). The 'Params' tab is selected, showing query parameters: start\_date (2018-11-01Txxxxxx), end\_date (2020-01-01Txxxxxx), location (US), and key\_terms (mers-cov). The 'Headers' tab shows '(6)' headers. The 'Body' tab is selected, showing a JSON response structure:

```
7 |   "reports": [
8 |     {
9 |       "diseases": [
10 |         "mers-cov"
11 |       ],
12 |       "event_date": "2009-12-xx xx:xx:xx to 2009-12-xx xx:xx:xx",
13 |       "locations": [
14 |         {
15 |           "geonames_id": 4796512
16 |         },
17 |         {
18 |           "geonames_id": 12217089
19 |         },
20 |         {
21 |           "geonames_id": 2193733
22 |         },
23 |         {
24 |           "geonames_id": 3573703
25 |         }
26 |       ]
27 |     }
28 |   ]
29 | }
```

Below the body, status information is displayed: Status: 200 OK, Time: 4.63 s, Size: 52.02 KB, and a 'Save Response' button. At the bottom, there are buttons for 'Pretty', 'Raw', 'Preview', 'Visualize', 'JSON', 'Find and Replace', 'Console', 'Bootcamp', 'Build', 'Browse', and help icons.

```
~/3011/SENG3011_2000K/PHASE_1/TestScripts(combined*) » python3 getLocation.py 4796512
St Croix
(base) --
```



## Saint Croix

Island in the Caribbean Sea

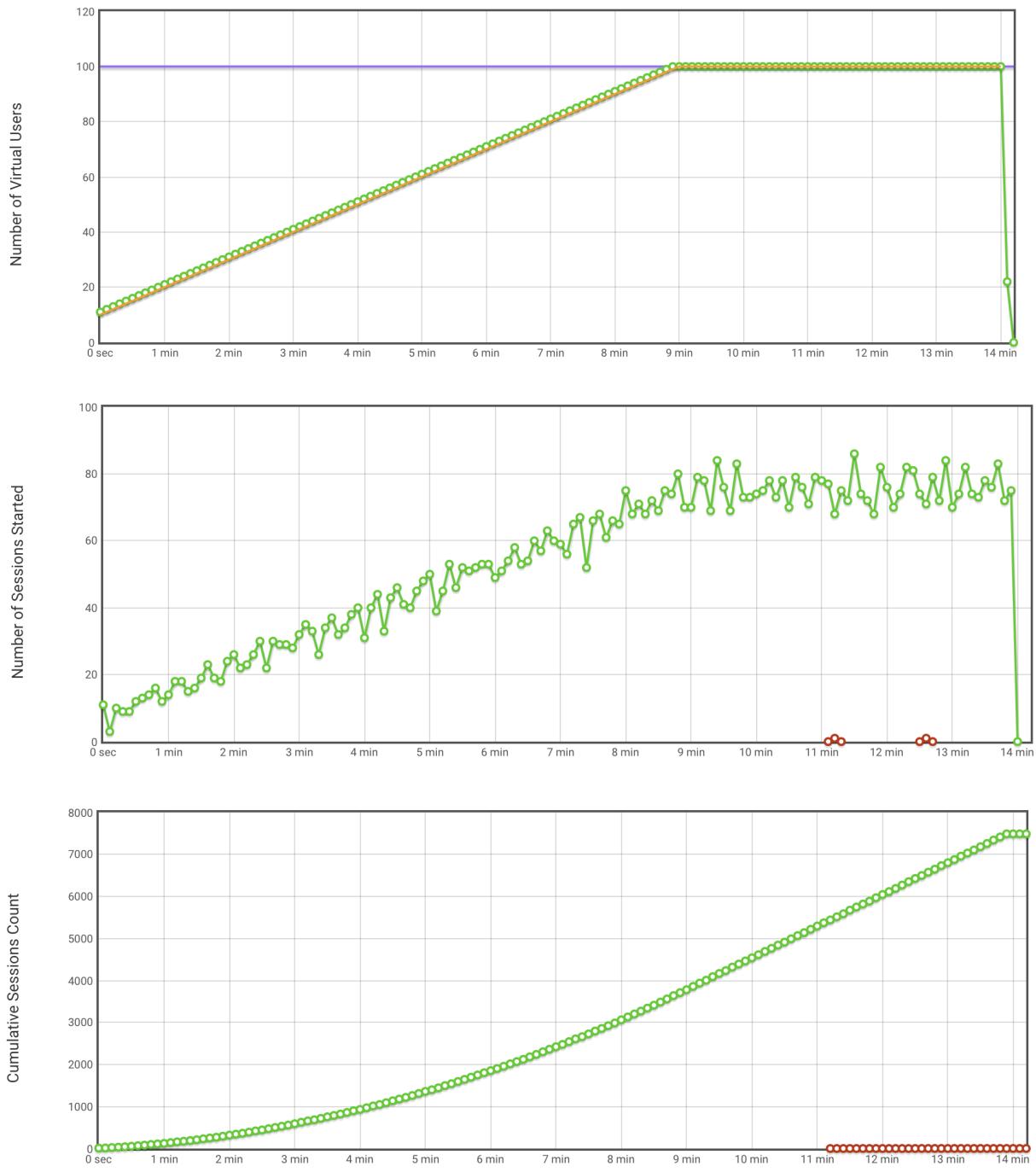
4.7 ★★★★★ 662 Google reviews

St. Croix is one of the U.S. Virgin Islands, in the Caribbean. Bright yellow Fort Christiansvaern is among Christiansted National Historic Site's Danish colonial buildings. West, St. George Village Botanical Gardens occupies an old sugar plantation. Point Udall's Millennium Monument marks the easternmost point of U.S. territory. North, Salt River Bay National Park has archaeological sites, mangroves and coral reefs. — Google

## Performance Test

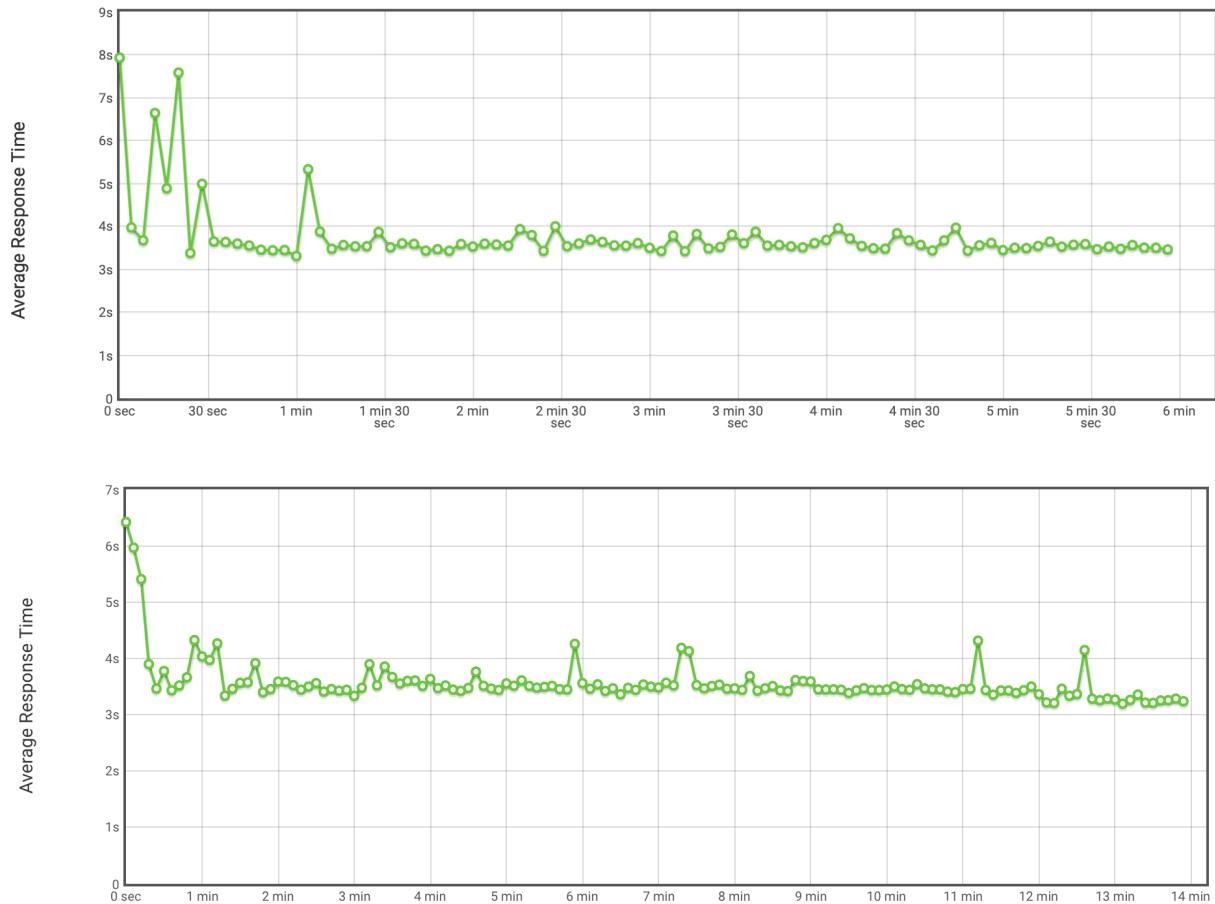
### Load Test

We use LoadView to conduct the load test. We tried different scenarios. The initial scenario started with 5 users and raised the number of users generally up to 50. The total number of sessions are about 1200 during this test. The API works well and no error occurs. Then we started with 10 users and gradually raised the number of users to 100. And we kept 100 users for 5 mins to verify if the API can stably handle this. The total number of sessions is about 7500. The API works well overall, although there are a few errors. As a result, we claim that our API can handle at least 100 users simultaneously. Below is the result for the second scenario.



## Speed Test

The speed tests are also conducted using LoadView. During the process for testing load, we also monitored the average response time. The result is shown below.



It can be seen that the average response time is around 3.5s, which is Acceptable.

## Security Test

For the convenience of users, we allow both HTTP and HTTPS requests to access our resources. And we also do not require user authentication.

Besides these, all other security aspects are ensured by the third party platform. We use Google Cloud Function to deploy the API and Google Firebase to manage the data. Google Cloud Platform has a number of security services. For example, it has built-in protection to protect the application, and the transit of data is encrypted. So we trust that the API is security based on the service of Google.