Testing Documentation

1 Testing Process

1.1 Running the Tests

Our tests are managed in Postman, an automated testing client. Postman allows us to simply and clearly create tests by specifying a URL, request type and set of input parameters, and the expected output of the response in JSON form. Using the client, we can specify one or more tests to run, and the number of iterations for which we should run them, allowing us to ensure that our API behaves consistently under repeated calls. It also provides us with the time and size of each response, which we can use to estimate the API's performance.

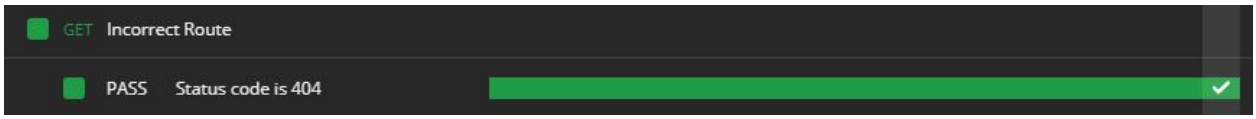
We made 2 collections of tests in Postman. The first collection tests a locally running API on localhost:5000, so that we can quickly run tests on development code. The second collection runs the same tests on the deployed API, so that we can test and monitor the status of the publicly accessible system.

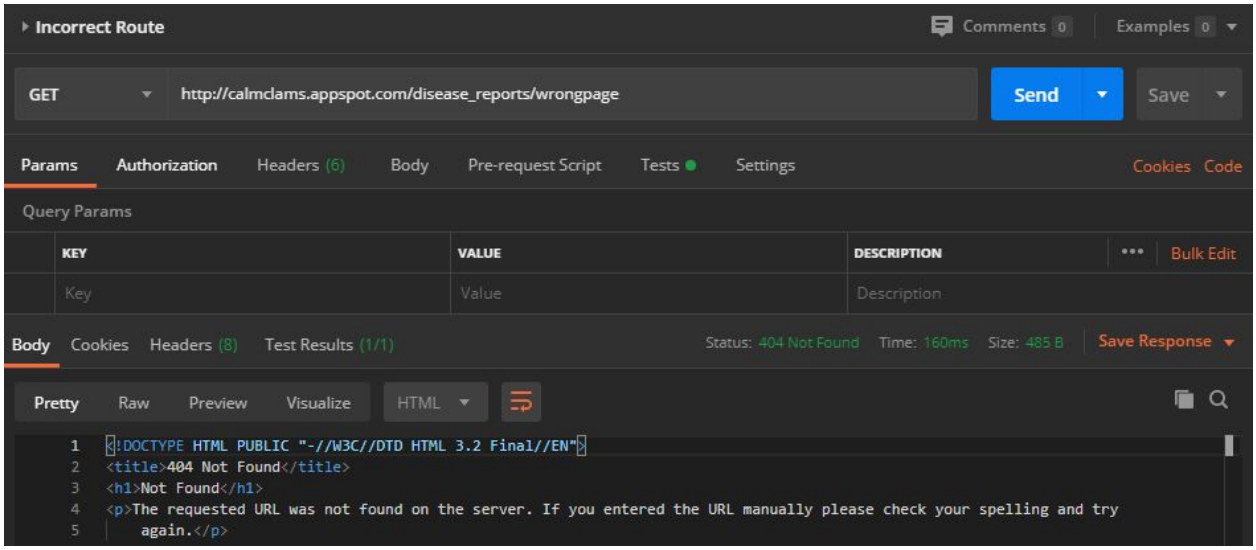
Once exported from Postman, the tests can also be run through the command line using the "newman" command, making it easier for developers to quickly run a suite of tests after making local changes.

1.2 Test Cases

Incorrect Route

Input	Invalid URL.
Expected response	404 Not Found, message to user that the URL cannot be found.
Purpose of test	The API should return a 404 response for URLs other than the /disease_reports path used to search for reports.
Response time	~30 ms

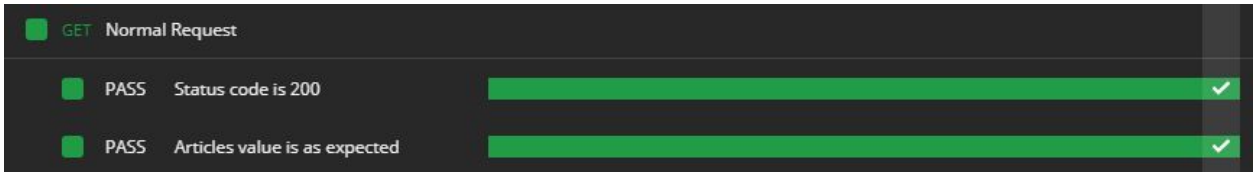


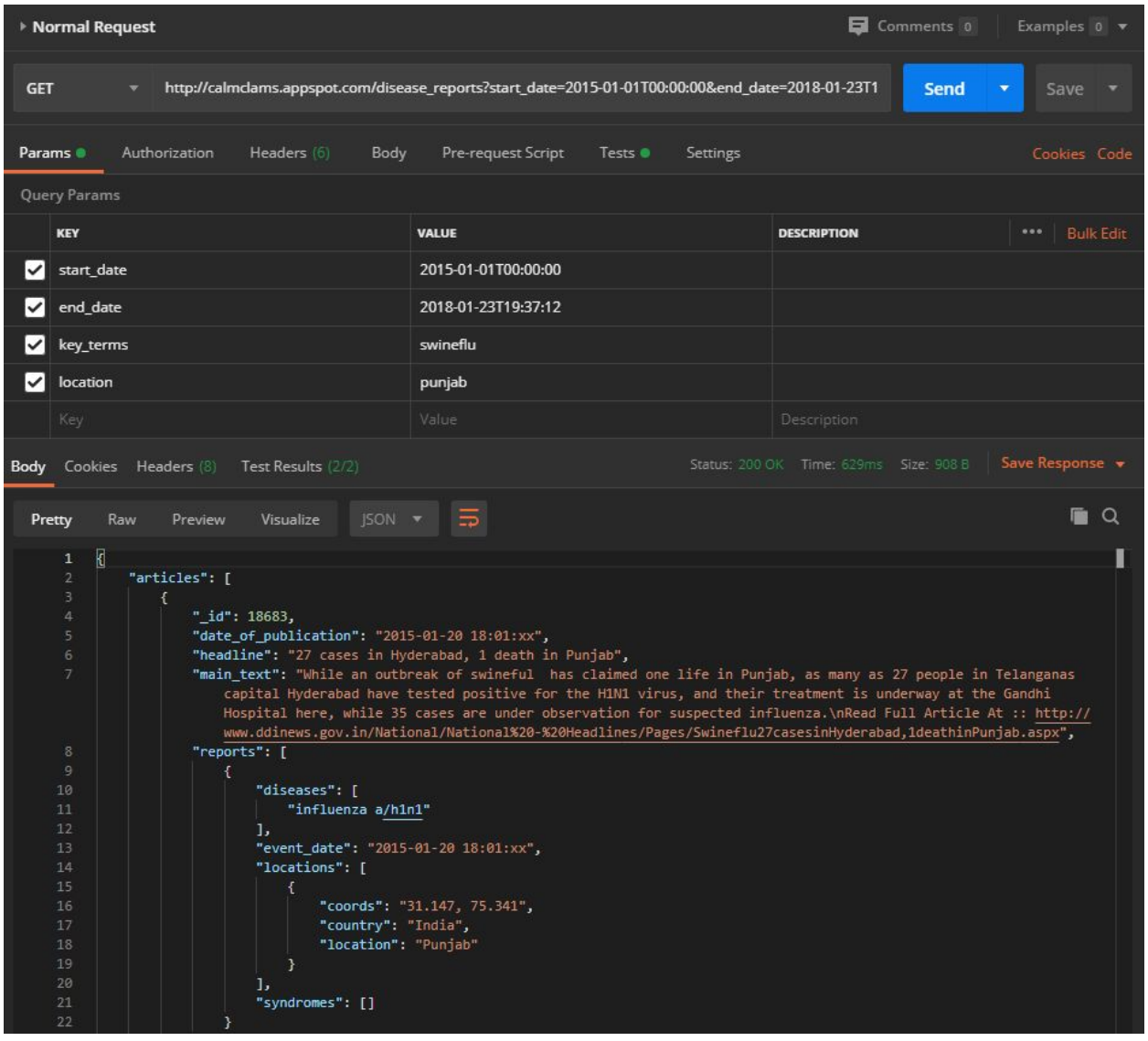


Normal Request

Input	Valid start and end dates, a single key term, a valid location.
Expected response	200 OK, JSON object containing a list of articles matching the specified search criteria.

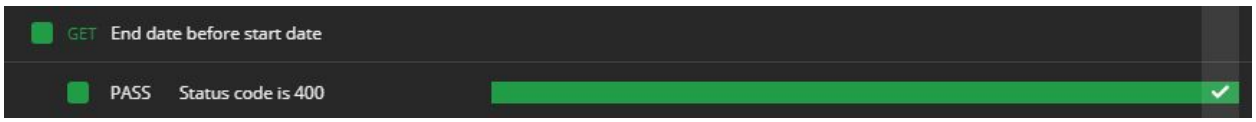
Purpose of test	Given a set of valid inputs, if the API has data matching the search criteria, it should return a JSON object with a consistent form to the user with the data and details of the request made.
Response time	~500ms

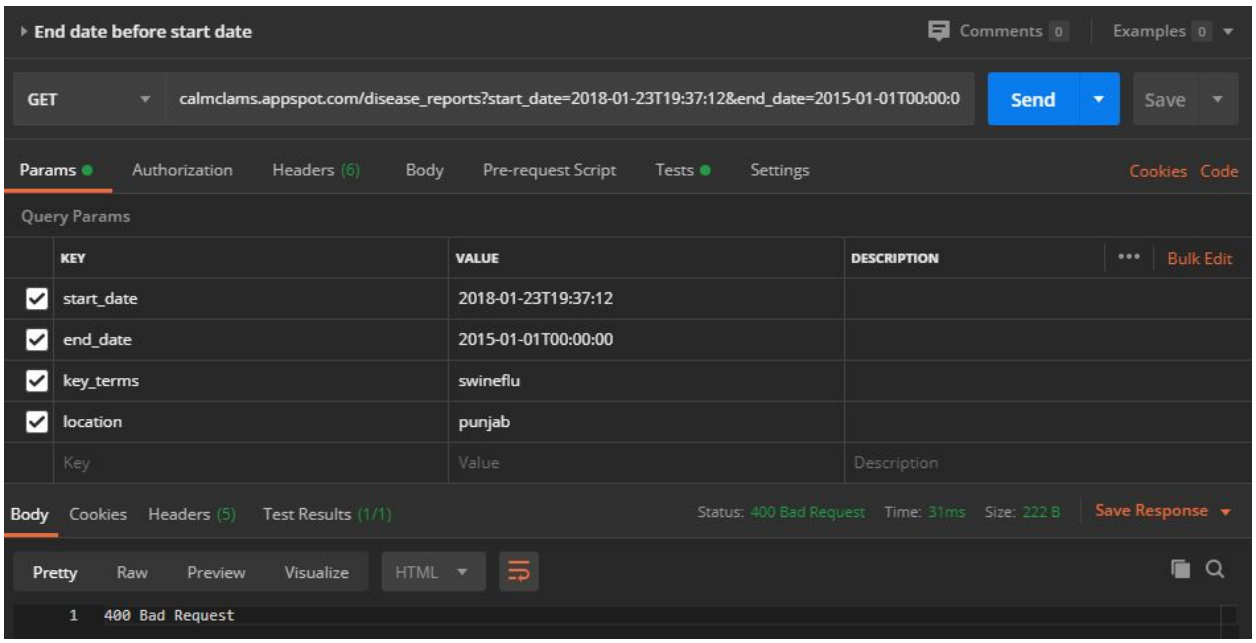




End Date Before Start Date

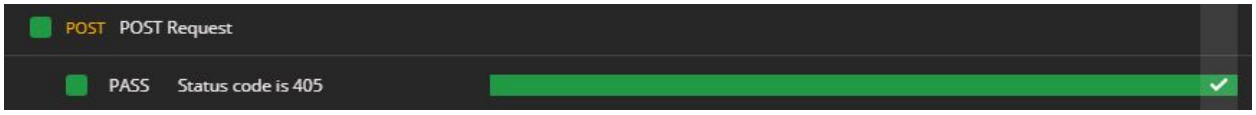
Input	A start date chronologically earlier than an end date, with otherwise valid input data.
Expected response	400 Bad Request
Purpose of test	The API should return a 400 response for user data that doesn't match the required specifications, to indicate that the input was invalid.
Response time	~40ms

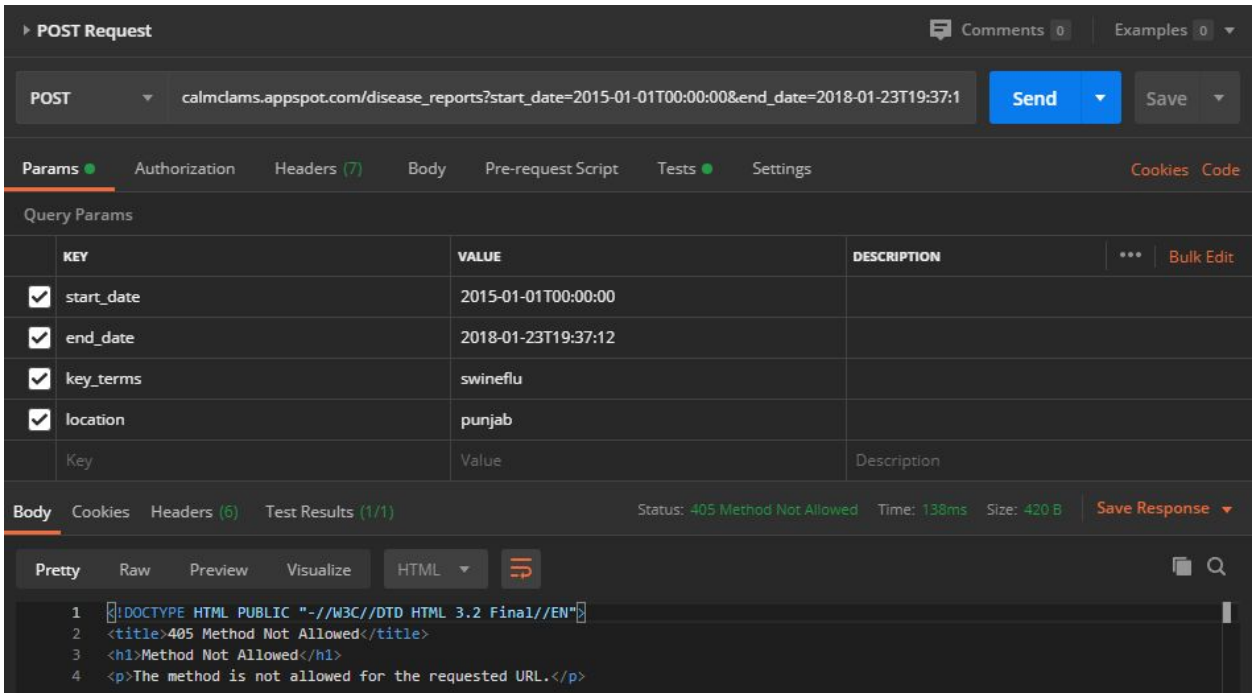




POST Request

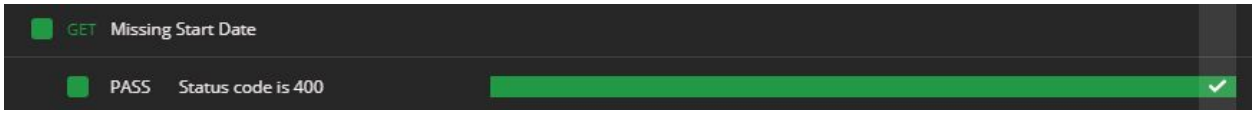
Input	A POST Request with otherwise valid input data.
Expected response	405 Method Not Allowed
Purpose of test	The API accepts queries through GET requests, and should indicate to the user that their request type was invalid for other requests, regardless of the correctness of other input data.
Response time	~30ms

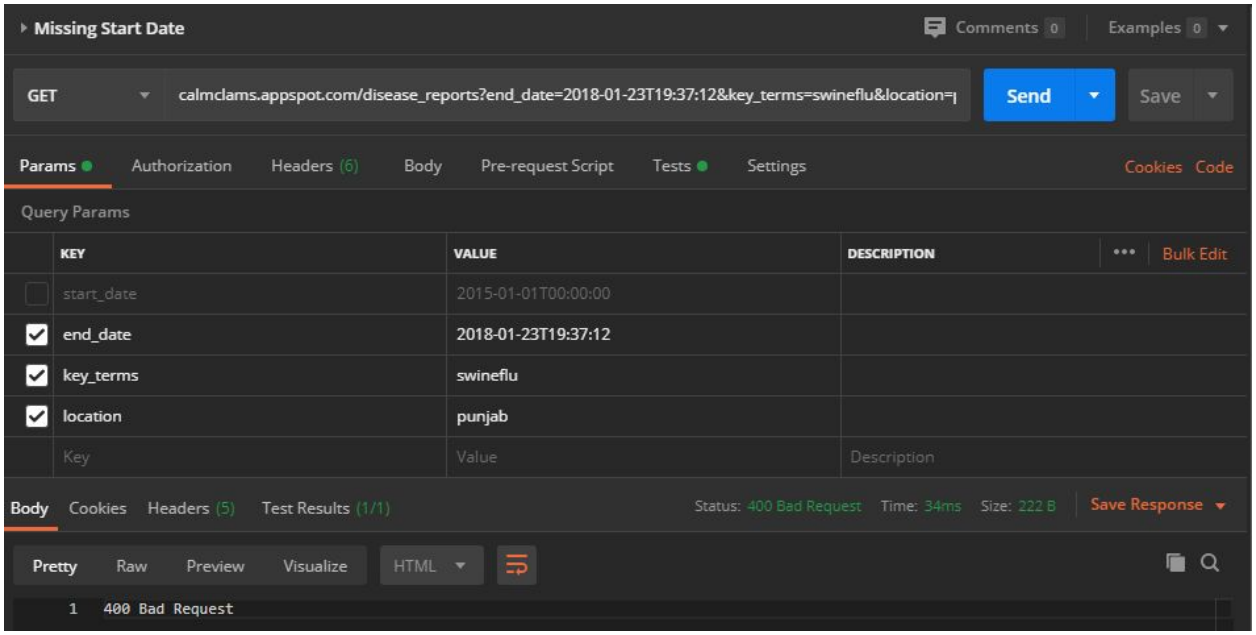




Missing Start Date

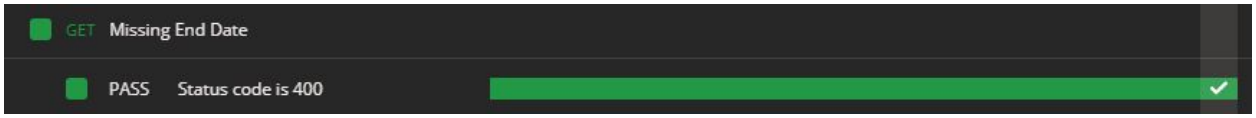
Input	Valid input data with the start date missing.
Expected response	400 Bad Request
Purpose of test	The API should return a 400 response for user data that doesn't match the required specifications, to indicate that the input was invalid. The start date is a required input, so any request in which it is missing is invalid.
Response time	~40ms

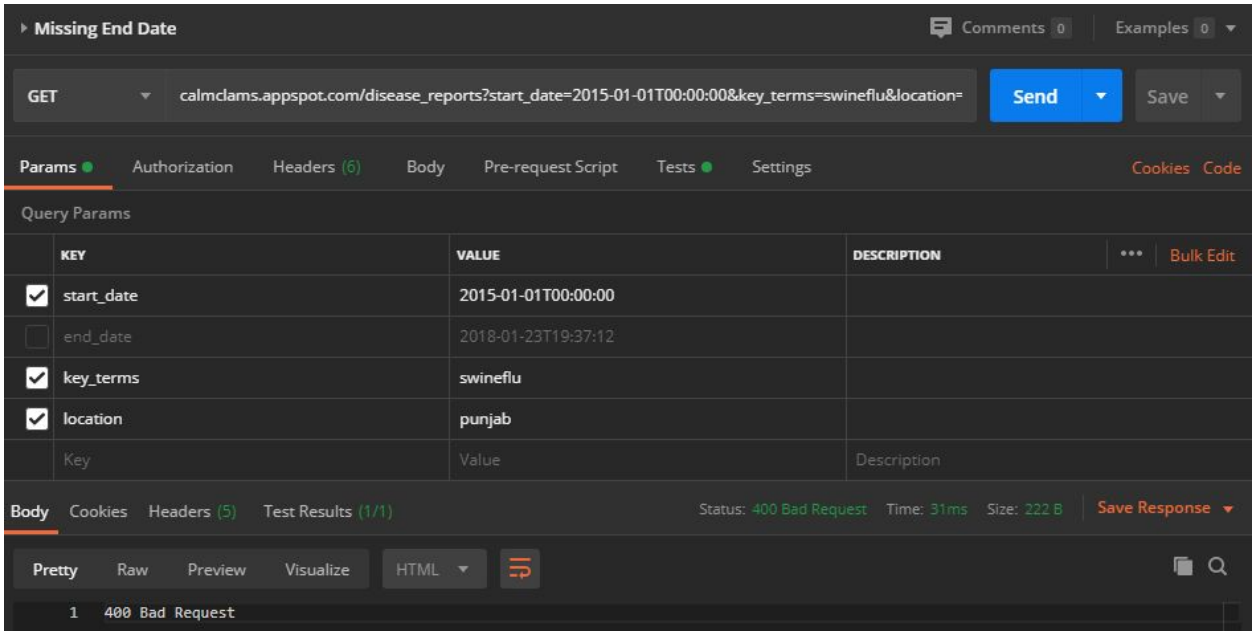




Missing End Date

Input	Valid input data with the end date missing.
Expected response	400 Bad Request
Purpose of test	The API should return a 400 response for user data that doesn't match the required specifications, to indicate that the input was invalid. The end date is a required input, so any request in which it is missing is invalid.
Response time	~40ms

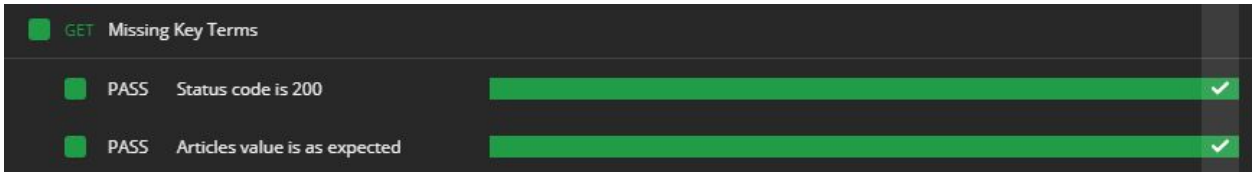


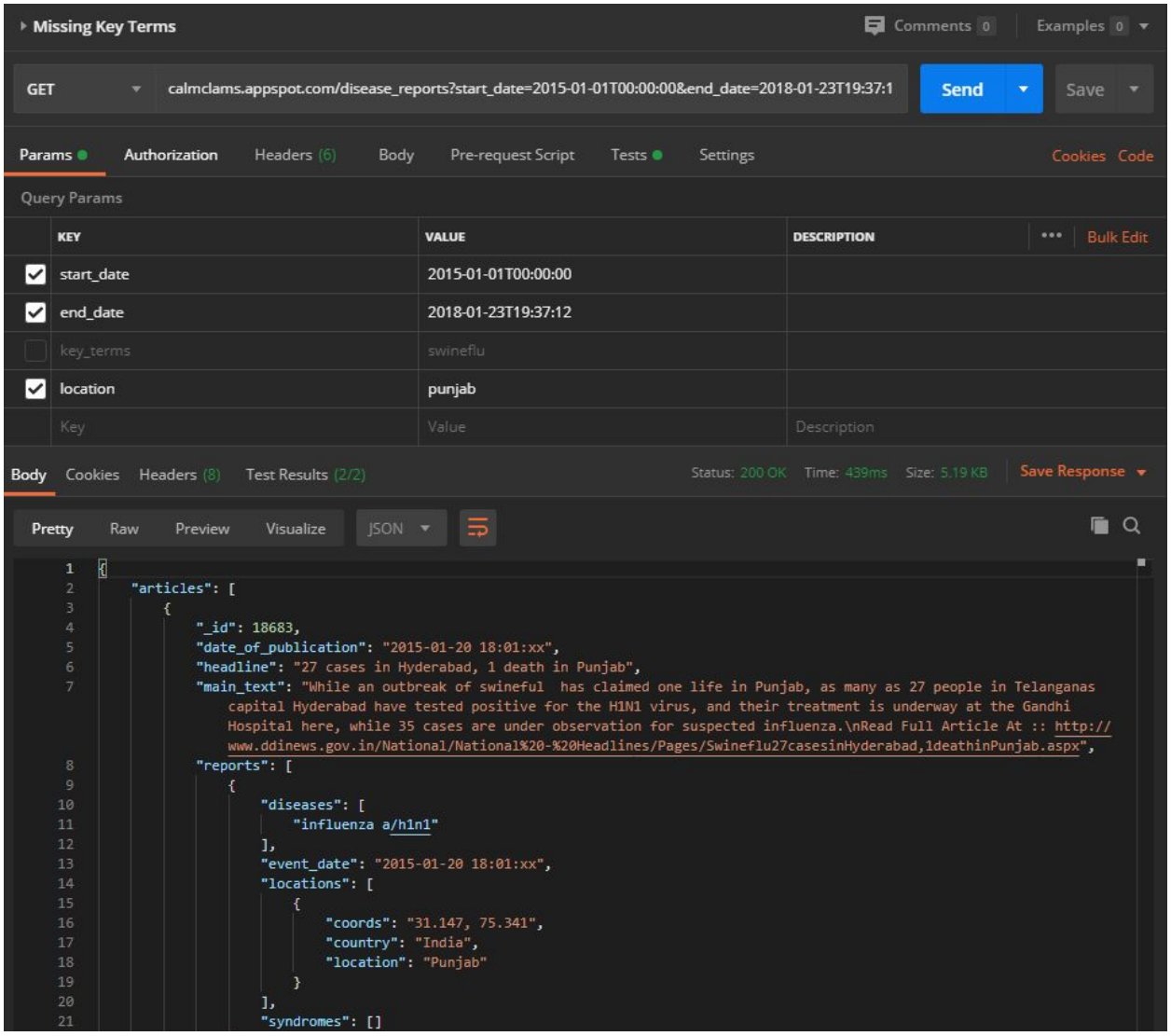


Missing Key Terms

Input	Valid start and end dates, no key term, a valid location.
Expected response	200 OK, JSON object containing a list of all articles between the start and end dates for the given location.

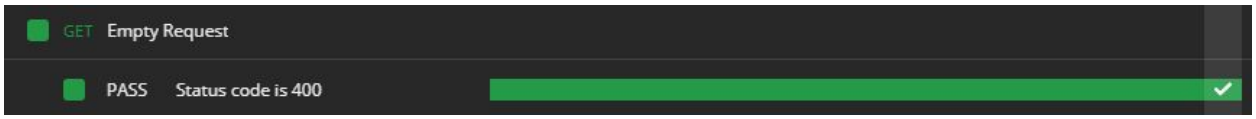
Purpose of test	Given a set of valid inputs, if the API has data matching the search criteria, it should return a JSON object with a consistent form to the user with the data and details of the request made. When no key terms are provided, all records for the given time period and location are matched.
Response time	~400ms

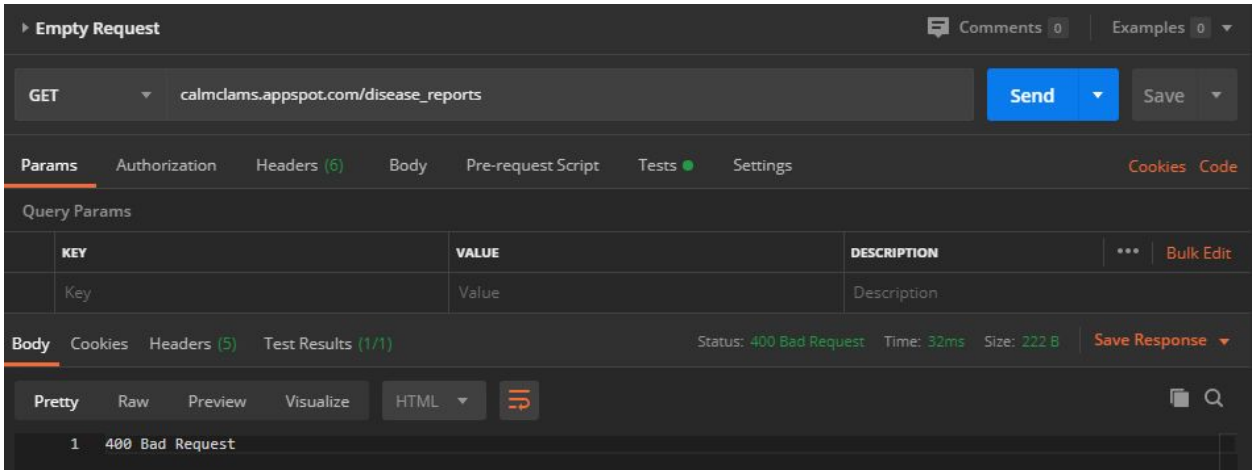




Empty Request

Input	A GET Request with no other input data
Expected response	400 Bad Request
Purpose of test	The API should return a 400 response for user data that doesn't match the required specifications, to indicate that the input was invalid. The start date, end date and location parameters are all required, so any request where they are missing is invalid.
Response time	~30ms

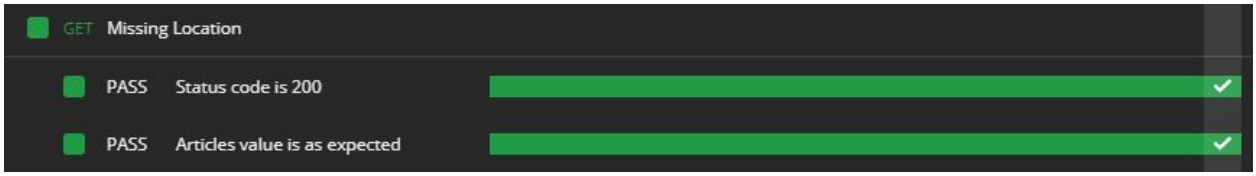


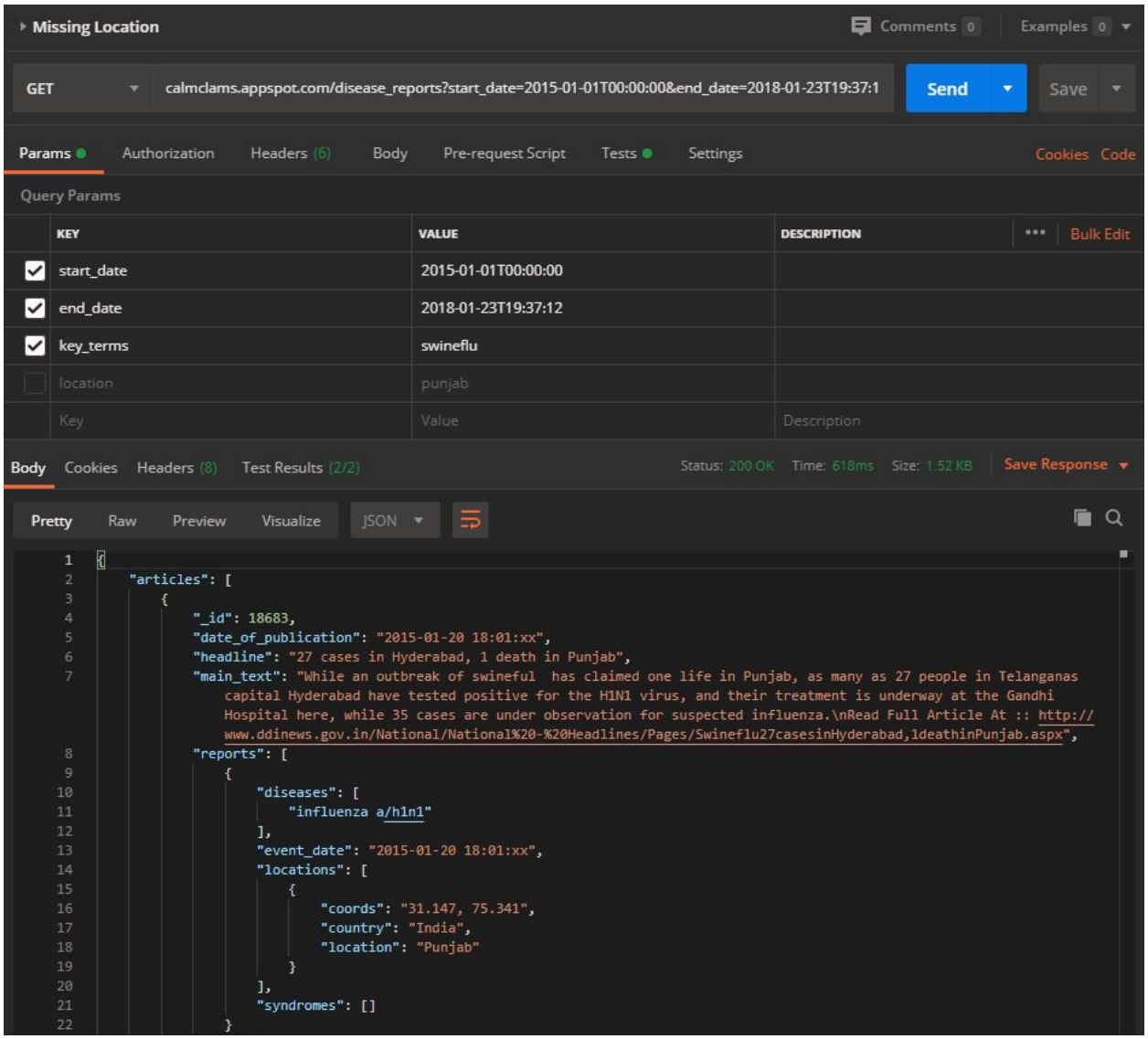


Missing Location

Input	Valid start and end dates, valid key term, no location.
Expected response	200 OK, JSON object containing a list of all articles between the start and end dates for the given key terms.
Purpose of test	Given a set of valid inputs, if the API has data matching the search criteria, it should return a JSON object with a consistent form to the

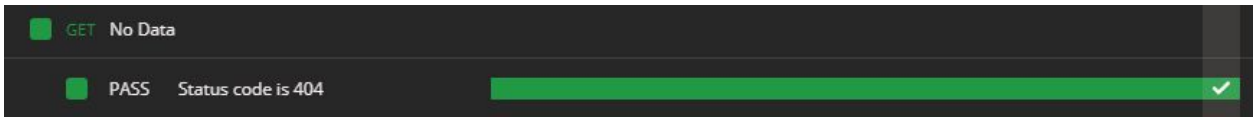
	user with the data and details of the request made. When no location is provided, all records for the given time period and key terms are matched.
Response time	~500ms

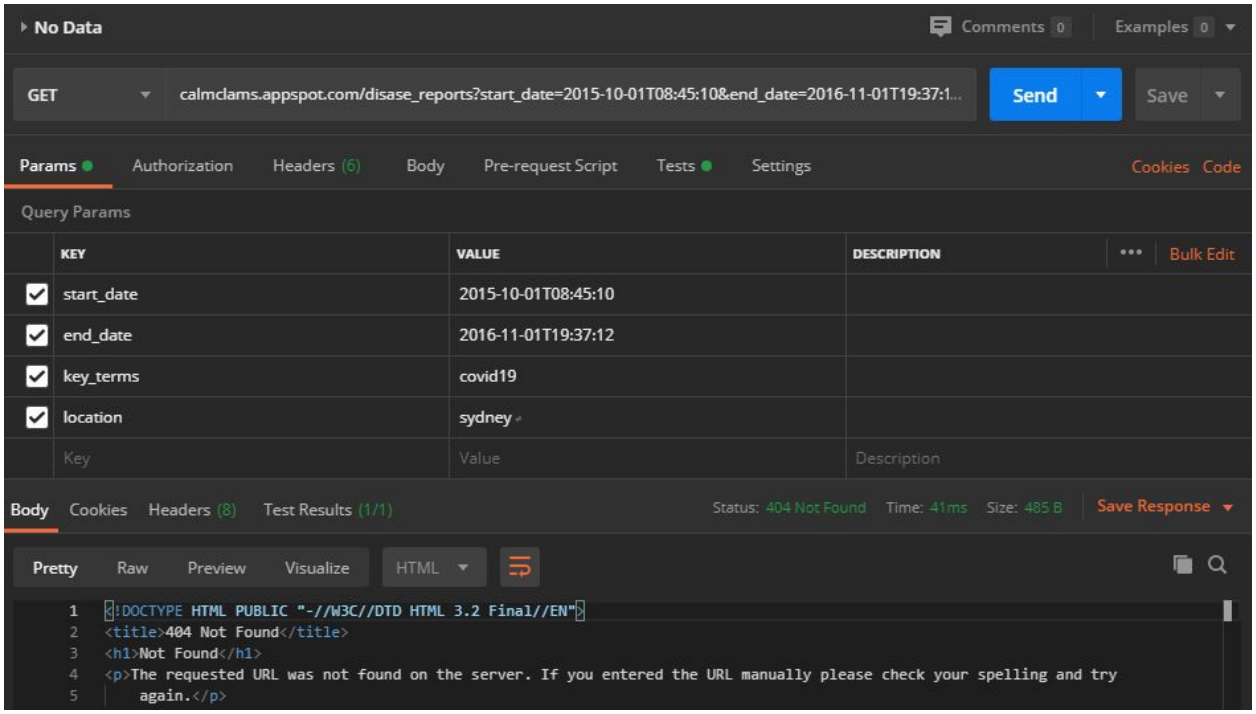




No Data

Input	Valid start and end dates, a single key term, a valid location, but no matching articles.
Expected response	404 Not Found
Purpose of test	Given a set of valid inputs, if the API cannot find any data matching the search criteria, it should return a 404 response to the user to indicate that data matching their inputs cannot be found.
Response time	~40ms

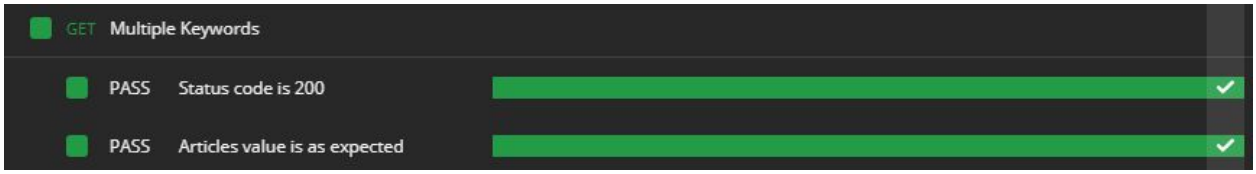


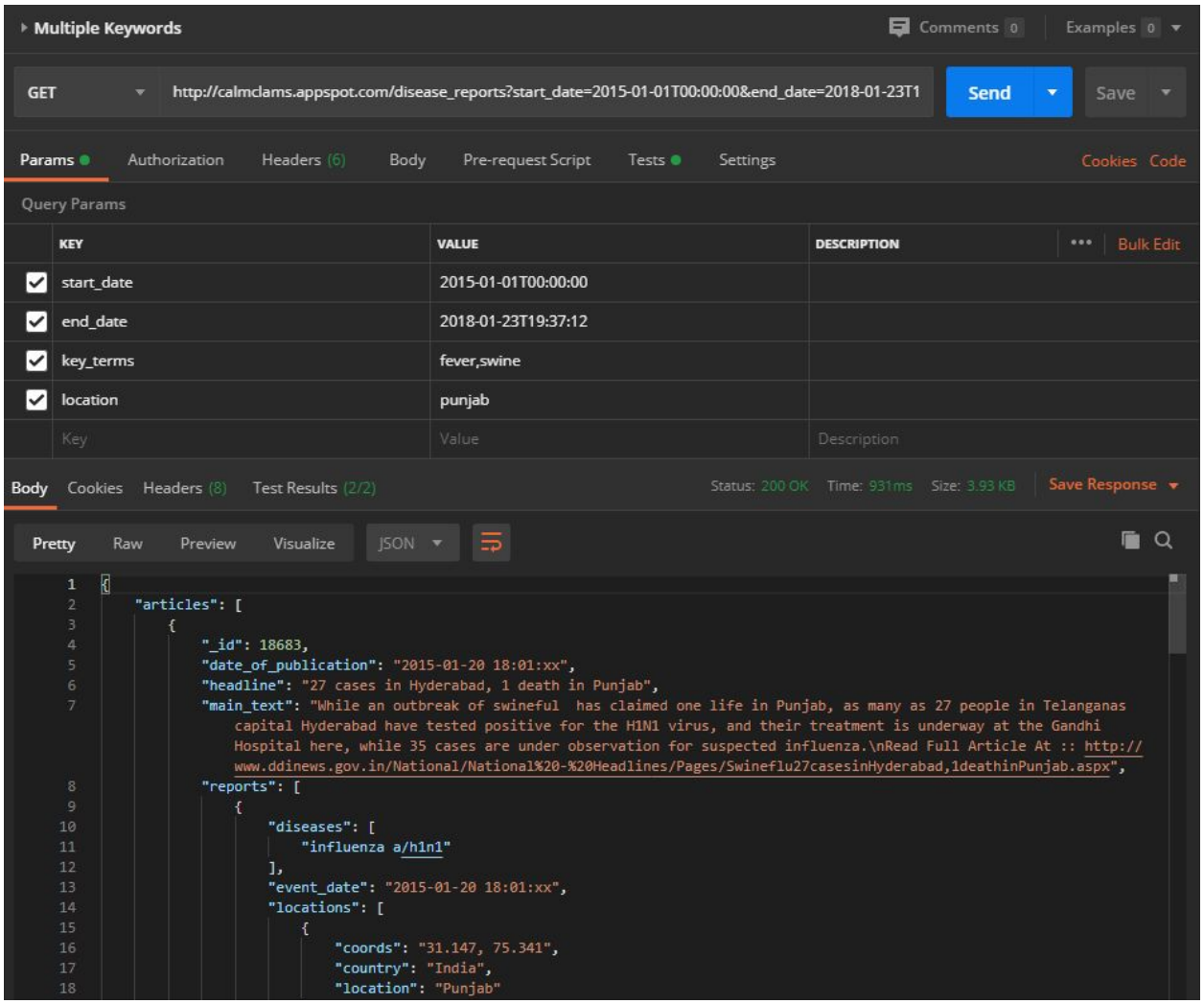


Multiple Keywords

Input	Valid start and end dates, multiple valid key terms, valid location.

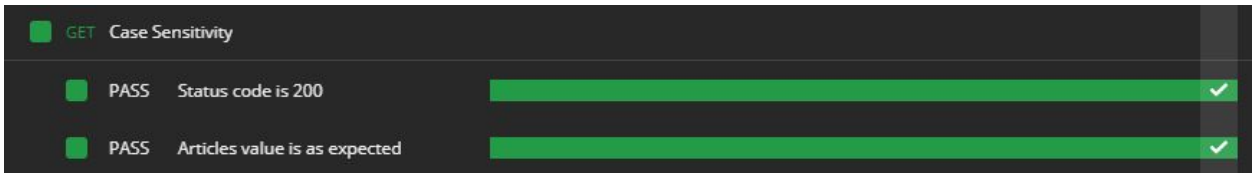
Expected response	200 OK, JSON object containing a list of all articles between the start and end dates for the given location matching at least one key term.
Purpose of test	Given a set of valid inputs, if the API has data matching the search criteria, it should return a JSON object with a consistent form to the user with the data and details of the request made. When multiple key terms are provided, any article containing at least one of the key terms should be matched.
Response time	~600ms

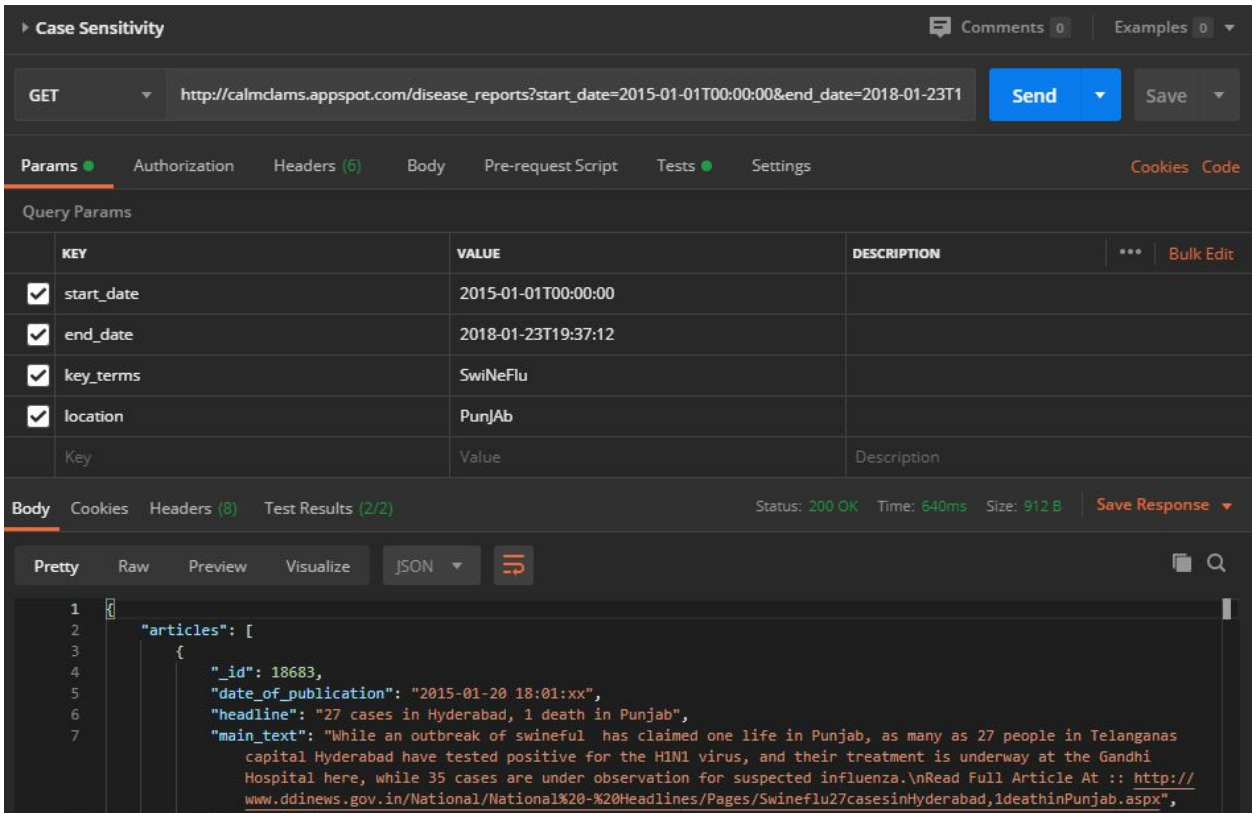




Case Sensitivity

Input	Valid start and end dates, valid key term and location with unusual capitalisation.
Expected response	200 OK, JSON object containing a list of all articles between the start and end dates for the given key term and location, matched case-insensitively.
Purpose of test	Given a set of valid inputs, if the API has data matching the search criteria, it should return a JSON object with a consistent form to the user with the data and details of the request made. Location and key term matches should be performed case-insensitively.
Response time	~500ms





2 Testing Environment

The testing environment we used is coupled with the tool that we used, being Postman. Postman allows for quick creation of requests, with multiple tests running on those requests. Through this we were able to create a comprehensive test suite, with all possible cases of requests that can be made to our API and a variety of tests performed on each case. These tests checked data such as the response code and the body of the response to check that it is correct.

Also, Postman allows for us to have a team workspace, which means we can collaborate on the creation of the tests. If anyone wanted to add extra tests to the test suite they would change the collections in the team workspace to include those tests. They would also need to export the collections to the GitHub, as there is no way to sync between exported collections and online collections.

3 Limitations

3.1 Security

We do have some limitations in our testing suite however, due to difficulty of creating the tests. We were unable to perform any security related tests, such as penetration testing, as none of us have an understanding of how to perform automated penetration tests. If this API was to go public and be used outside of this university course we would invest some time in making sure our API has rigorous security measures.

In saying that, as our API is only serving data and not recording any user related state, we do not perform any authentication. So the need for security in this API is low.

3.2 Performance

We were also unable to check the performance of our API within the automated test suite for various factors. One of those factors is that the time it takes to send the response is dependent on how much data the user has requested. If we had implemented some kind of pagination into our API then this might have been possible, but due to time constraints we were unable to add that.

Also, with the constantly updating data from the web scraper the size of the response may change, meaning that we would be unable to predict what the performance would be like.

Despite this, we are able to test the performance manually through Postman, as postman is able to tell us the response time of the request. Ideally this would be automated as the rest of our tests are, but manual testing will at least be able to ensure that our API responds in a timely manner.