





SOFTWARE TOOL ARTICLE

NaijaCovidAPI: an application programming interface for retrieval of COVID19 data from the Nigerian Center for Disease Control web platform

[version 1; peer review: 2 approved with reservations, 1 not approved]

Emmanuel Baldwin Mbaya ¹⁻³, Babatunde Alao^{1,2}, Philip Ewejobi^{1,2}, Innocent Nwokolo^{1,2}, Victoria Oguntosin ¹, Emmanuel Adetiba^{1,2,4}

¹Department of Electrical & Information Engineering, Covenant University, Ota, Ogun State, Nigeria

²Covenant Applied Informatics and Communication Africa Center of Excellence, Covenant University, Ota, Ogun State, Nigeria

³Department of Computer Science, Federal University Gashua, Gashua, Yobe State, Nigeria

⁴HRA, Institute for Systems Science, Durban University of Technology, Durban, South Africa

V1 First published: 02 Dec 2021, 10:1227
<https://doi.org/10.12688/f1000research.74998.1>
 Latest published: 02 Dec 2021, 10:1227
<https://doi.org/10.12688/f1000research.74998.1>

Abstract

Background: In this work, a COVID19 Application Programming Interface (API) was built using the Representational State Transfer (REST) API architecture and it is designed to fetch data daily from the Nigerian Center for Disease Control (NCDC) website.




Methods: The API is developed using *ASP.NET Core* Web API framework using C# programming language and Visual Studio 2019 as the Integrated Development Environment (IDE). The application has been deployed to Microsoft Azure as the cloud hosting platform and to successfully get new data from the NCDC website using Hangfire where a job has been scheduled to run every 12:30 pm (GMT + 1) and load the fetched data into our database. Various API Endpoints are defined to interact with the system and get data as needed, data can be fetched from a single state by name, all states on a particular day or over a range of days, etc.



Results: The results from the data showed that Lagos and Abuja FCT in Nigeria were the hardest-hit states in terms of Total Confirmed cases while Lagos and Edo states had the highest death casualties with 465 and 186 as of August 2020. This analysis and many more can be easily made as a result of this API we have created that warehouses all COVID19 Data as presented by the NCDC since the first contracted case on February 29, 2020. This system was tested on the BlazeMeter platform, and it had an average of 11Hits/s with a response time of 2905milliseconds.

Conclusions: The extension of NaijaCovidAPI over existing COVID19 APIs for Nigeria is the access and retrieval of previous data. Our contribution to the body of knowledge is the creation of a data hub for Nigeria's COVID-19 incidence from February 29, 2020, to date

Open Peer Review

Approval Status   

	1	2	3
version 1			
02 Dec 2021	view	view	view

1. **Xiaojin Li**, The University of Texas Health Science Center at Houston, Houston, USA
2. **Micheal Olaolu Arowolo** , University of Missouri,, Columbia, USA
3. **Alejandro Zunino** , ISISTAN Research Institute, CONICET-UNICEN, Buenos Aires, Argentina

Any reports and responses or comments on the article can be found at the end of the article.

Keywords

COVID19 API, NCDC, REST architecture, ASPNET Core Web



This article is included in the **Emerging Diseases** and **Outbreaks** gateway.

Corresponding author: Victoria Oguntosin (victoria.oguntosin@covenantuniversity.edu.ng)

Author roles: **Mbaya EB:** Data Curation, Formal Analysis, Investigation, Methodology, Resources, Software, Writing – Original Draft Preparation, Writing – Review & Editing; **Alao B:** Formal Analysis, Investigation, Writing – Original Draft Preparation; **Ewejobi P:** Investigation, Methodology, Visualization; **Nwokolo I:** Investigation, Methodology, Validation; **Oguntosin V:** Conceptualization, Project Administration, Supervision, Writing – Review & Editing; **Adetiba E:** Project Administration, Resources, Software, Supervision, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: The author(s) declared that no grants were involved in supporting this work.

Copyright: © 2021 Mbaya EB *et al.* This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Mbaya EB, Alao B, Ewejobi P *et al.* **NaijaCovidAPI: an application programming interface for retrieval of COVID19 data from the Nigerian Center for Disease Control web platform [version 1; peer review: 2 approved with reservations, 1 not approved]** F1000Research 2021, **10**:1227 <https://doi.org/10.12688/f1000research.74998.1>

First published: 02 Dec 2021, **10**:1227 <https://doi.org/10.12688/f1000research.74998.1>

Introduction

The Coronavirus disease 2019, known as COVID-19, was termed by the team of Taxonomy of Viruses as SARS-COV-2.¹ The first case occurred in December 2019 in a place known as WUHAN at the Hubei province in China.^{2,3} Like a wildfire, it grew and reached 233 countries with millions of confirmed cases and deaths.⁴

Nigeria recorded its first case on February 28, 2020, when an Italian man working in Nigeria returned from Italy to Lagos. This was confirmed by the Lagos University Teaching Hospital.⁵ Since the incidence of the index case, the number of deaths, confirmed, active and discharged cases, has been on the increase in Nigeria and the rest of the world.⁶

Data has become increasingly important in research to produce discoveries with the introduction of computers, which brought with them computing, simulation, and modelling. Furthermore, because most of the research on COVID-19 are funded by national governments, the data and outcomes are publicly available.⁷⁻⁹

The World Health Organization (WHO) has designed a dashboard that gives real-time data about the global statistics of the COVID-19 outbreak. In addition, Coronatracker was developed to provide real-time news, statistics and to curb the spread of fake information about the disease.^{4,10} For instance, in Nigeria, the Presidential Task Force (PTF) on COVID-19 gave daily updates from the case's onset for about 8 months. The NCDC has developed a platform to provide information in respect of the daily incidences on its website and other social media platforms.⁶ These platforms were developed to provide real-time data for public health and research purposes. To analyze research datasets, Python has continued to increase its popularity amongst researchers in scientific domains where MATLAB and SPSS previously dominated. Most often, Python's Panda's library uses Comma Separated Values (CSV) file format to provide data for downloads from online platforms.¹¹ Another alternative is to provide an Application Programming Interface (API) for users to have access to data from web portals.

API has existed since the advent of the Internet for the exchange of data between two or more programs. It has gotten a high level of acceptance recently that "pundits" are saying we are in the API economy. This position might likely be because people are now more interconnected through different types of applications like never, and individuals are now demanding online services at a higher rate than ever. Notably, API has now made it possible for businesses to connect for enhanced capacity and profitability far beyond what has been known historically.¹²

Big corporations such as Uber, Airbnb, PayPal, and a host of others depend on several APIs and developer services, with Uber paying Google the sum of \$58 million for its map services. Nigeria is not left behind in this race as several start-ups have jumped into the API economy with companies such as Paystack, who developed APIs to fast-track payments using Cards (credit and debit) and direct bank transfers.^{13,14}

In this paper, we report the development and deployment of a RESTful API to scrape COVID-19 data from the NCDC website and expose its response in JavaScript Object Notation (JSON) format. We built a web scraper that scrapes the daily updates from the NCDC website and archives them on a cloud database. Furthermore, we developed a Web API that allows users/developers to consume the data in JSON format.

The rest of this paper contains a literature review, the methods, the results, and the conclusions.

Literature review

Application Programming Interface (API) is the intermediary between software; it states how the interaction will be as well as the data formats.¹⁴ This section focuses on the previous works done by scholars concerning different application domains.

A RESTful API developed using JavaScript; scrapes data from the NCDC website and exposes its response in JSON. The application gives real-time data as it appears on the NCDC website with details across each state; details such as deaths, confirmed cases, cases on admission, discharged and the total number for each segregated data at the national level.^{6,15}

Breast cancer is the second leading cause of mortality among women worldwide; nevertheless, a tiny proportion of men are also at risk. Early detection of this condition is critical since therapy can then be started, enhancing the chance of survival. The authors created an application that can assist health professionals in the prediction and complement the detection of breast cancer. This prediction will be accomplished by training a model using Google Prediction API. The API has integration with various programming languages such as Java, GO, JavaScript, Python, PHP and Ruby.¹⁶

Researchers, physicians, and the general public, more than ever, face a massive challenge in keeping up with the rapid pace of findings and advancements of the influx of clinical trials. To address this issue, researchers combed the [ClinicalTrials.gov](https://clinicaltrials.gov) database for COVID-19-related clinical trials, created unique reports to summarize results, and made meta-data accessible through APIs.¹⁷

The research was carried out on characterizing illicit COVID-19 product sales. This involved collecting COVID-19-related data from Twitter and Instagram posts using a combination of site scraping on Instagram and filtering the public streaming Twitter API for keywords correlated with suspect COVID-19 marketing and sales. Data were analyzed using Natural Language Processing (NLP) and deep learning to classify possible buyers, who were then manually annotated for desired characteristics. They had used a personalized data dashboard to visualize illegal trading posts to provide public health information.¹⁸ Also, a reusable dataset using publicly accessible and crowd-verified data on the COVID-19 outbreak in Kerala from government bulletins and media outlets as part of a citizen science initiative via a front-end Web application and a JSON repository, which acts as an API for the front end, this was visualized and produced as a dashboard.¹⁹

The traditional APIs were previously built using Simple Object Access Protocol (SOAP) web service, an XML (Extensible Markup Language) based protocol for accessing web service over HTTP. This comes with some disadvantages that have been addressed in a new trend of APIs. API is now being migrated towards REST interface because of its simplicity, ability to support more straightforward programmatic access by returning either XML or JSON.²⁰ However, REST API is now becoming a new norm across various web services because of its improvement over performance, scalability and flexibility compared to SOAP services. Another advantage of REST that cannot be ignored over SOAP is that it consumes fewer resources, as well as its operation, do not include different standard and heterogeneous procedures that make it easier to tear domain and compose it when compared to SOAP.

It is understood that there is a significant difference in using SOAP web service and REST web devices. A web service was developed by TogoWS that explained these differences, an integrated web service that gives uniform access to database resources, parsers for database entries and converters for major data formats'.²¹ Web services can be categorized into data-retrieval services and analysis services. Both types of services can be exposed using either the REST or the SOAP architecture. In some cases, REST is better suited for data-retrieval services, and SOAP is more suitable for analysis services because REST is easily mapped to resource URIs. In contrast, SOAP usually requires a long execution time or complex parameters.²² Furthermore, SOAP architecture brings output in XML, while REST gives output in JSON in most cases.

From our extensive review, REST architecture is mostly utilized in the literature and is was adopted for this study. In the existing COVID API, gaps identified include the previous day's data are unavailable and so one cannot go back in time to assess the curve of the virus, data is retrieved in bulk, and one cannot select a particular state to inspect data for that state only. The work at hand overcomes these gaps by adding features that will make available useful data to researchers.

Methods

To realize the objective of this work, the following tools and frameworks were engaged to develop the web scraper, which fetches data from the NCDC website then the data is warehoused on MS SQL Server. Furthermore, a Web API that has endpoints for data to be consumed in JSON is developed.

1. ASP.NET Core Web API framework: This is a framework used for building HyperText Transfer Protocol (HTTP) services that are accessible from various platforms/clients including but not limited to web browsers, mobile devices, desktop applications etc.
2. Entity Framework (EF) Core: It is an Object and Relational Mapping (ORM) framework that supports *code first and database first* approach. It is an open-source, lightweight, extensible and cross-platform with support for several database engines. It is used to communicate with the database and EF Core makes it possible to carry out Create, Read, Update and Delete (CRUD) operations and much more without explicitly writing Structured Query Language (SQL) scripts.
3. HTML Agility Pack: It is an agile HTML parser that we used for the web scrapping component of the system. It supports XPath (XML Path Language) and uses it to build a read/write Document Object Model (DOM).
4. SQL Server Management Studio: This is an IDE for writing SQL queries and scripts; design, deploy and manage databases from MS SQL Server to Azure SQL Database.

5. Postman: It is an open-source tool for API testing, monitoring and publishing.
6. Swagger: this is an Interface Description Language (IDL) used for describing RESTful API. It helps to give a visual look at the various API endpoints and to execute the commands.
7. Hangfire: An easy way to perform background processing in .NET and .NET Core applications without the need of Windows Service or a separate process required.
8. Blazemeter: BlazeMeter is an online open source-based, enterprise-ready platform that unifies all the functionality needed in carrying out various tests such as functional testing, performance testing, API testing and much more.

The application block diagram is shown in **Figure 1**. The different procedures that were carried out to develop the application are hereafter described.

Web scraper

To web scrape a web page, one has to understand the Hypertext Mark-up Language (HTML) structure and then use the XPath of the element(s). The NCDC daily update has details for every state concerning their COVID cumulative data and the national summary of cases. **Figure 2** is a screenshot of the NCDC website showing the data that can be gotten from the webpage.

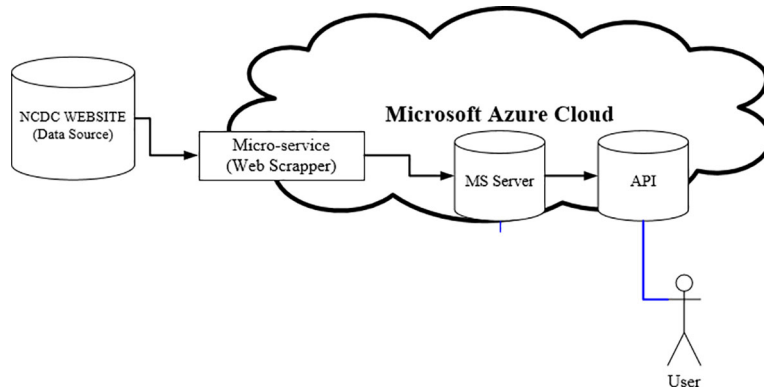


Figure 1. Application block diagram.

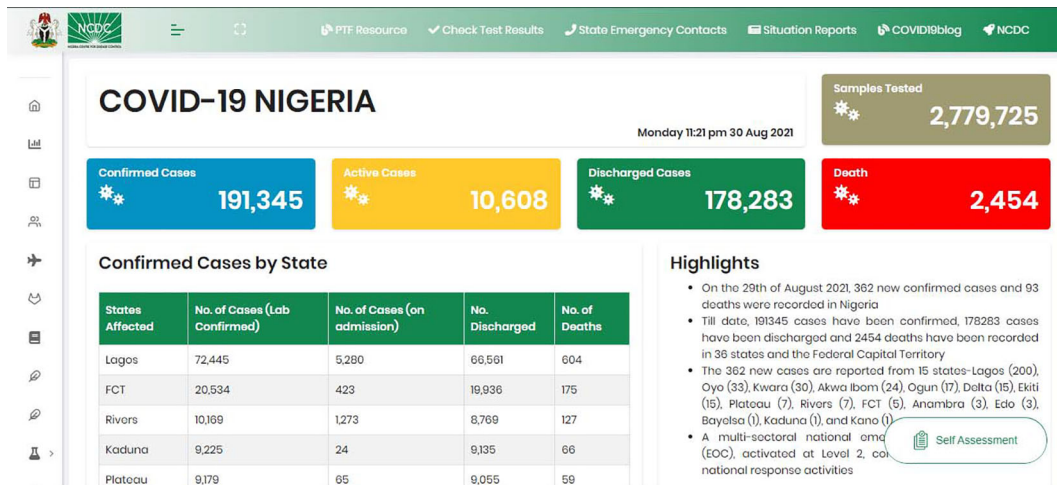


Figure 2. NCDC COVID-19 website.

Upon study of the source code of the NCDC website, the HTML elements containing the confirmed cases, active cases, deaths, and discharges have an XPath of “//h2[@class = 'text-right text-white’]” and that can be used to fetch all nodes that have such structure. First, the package was downloaded using NuGet Packages, then added to the project automatically by Visual Studio, and then imported into the class that will handle the scraping of data.

After data was scrapped, a validation was done to ensure that the date of the data was yesterday's date (this is because the real-time data on the NCDC website is the previous day's data). If the data validation failed, the scraping was rescheduled for another two hours, but if the validation passed, the data was stored in the database.

Figure 3 shows the Unified Modelling Language (UML) Class Diagram for this system. The *HomeController* is the class that handles all HTTP requests, and the various endpoints are represented in the form of methods. Annotation was used to state the HTTP method to respond and the GET method for all the endpoints.

The *DatabaseContext* class is the intermediary between the database and the system. The created tables are defined as properties with a returned type of *DbSet*. The *DatabaseContext* class inherits from the *DbContext* superclass and the superclass as a method of *OnModelCreating*, which is used to state the properties (such as Primary Key, Foreign Key, etc.) as defined in the tables.

DataService class implements the *IDataService* interface. It is good programming practice to use interfaces to model multiple inheritances, a feature that some object-oriented languages support that allows a class to have more than one superclass. The *DataService* type makes use of Language Integrated Query (LINQ) statements to execute queries on the database.

The *DataModel* is the model of data fetched from the NCDC website and the data model to be saved in the database. The *DataModel* class has the properties that capture the data needed from the NCDC website. Other details such as Sampled Cases, Confirmed Cases, Active Cases, Discharged, and Deaths at a national level are captured by the *ScrapeData()* method.

After data has been scrapped, a test was performed to ensure that the data fetched is the latest from the website. To achieve that, the date of the scrapped data is compared with the previous day's data (since that data in the NCDC website is a reflection of data in the last day) to ensure that it is not already captured. If the information does not pass this test, it will not be processed further, and then the routine is scheduled after two hours because it means the website has not been updated.

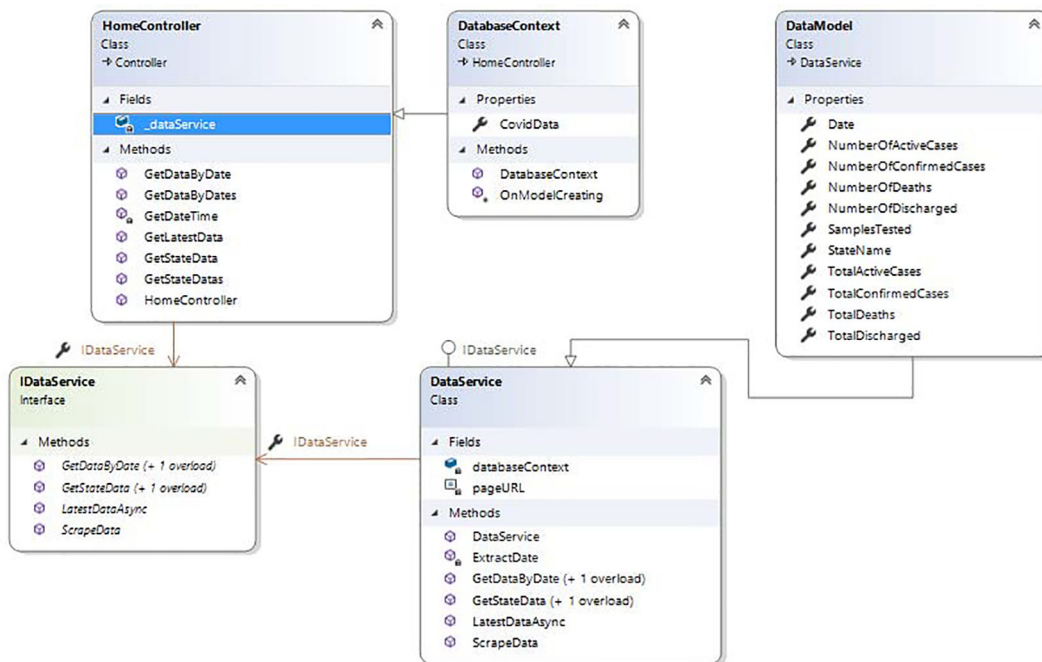


Figure 3. UML class diagram of the application.

If the data passes the test, the *SaveData()* method will be called to store the data in the database. Entity Framework was adopted to handle database transactions rather than writing Structured Query Language (SQL) scripts directly into the code.

Automation

Since data will need to be fetched from the NCDC website daily, the scrapping process was automated. Hangfire was adopted to achieve this automation as it is an open-source library that supports background jobs, which perform fire-and-forget, delayed and recurring, long-running, short-running, CPU or I/O intensive tasks inside ASP.NET applications and No Windows Service or Task Scheduler is required.

We utilized the delayed and recurring jobs option. Under the recurring option, the job was scheduled to run by 12:30 pm (GMT +1). This decision was made because the NCDC website is not normally updated in the early hours of the day. The delayed jobs option happen in instances where after data has been fetched but failed the date test (i.e., the data fetched has not been updated), it will be rescheduled to run after two hours.

The Hangfire package (*Hangfire.AspNetCore* and *Hangfire.SqlServer*) has to be installed from the NuGet Packages before it can be used in a project. After installation, it was registered with the Startup class in the ConfigureServices to use SQL Database using the already created connection while in the Configure method. Afterwards, it registered the Hangfire Dashboard and Server, which are additional services to monitor the background service and status.

The scraper is scheduled to run at 12:30 pm (GMT + 1), it uses what is known as cron expression. Cron expression uses five (5) asterisks (*) and asterisk mean every minute of every hour of every day of every month and every day of the week as shown in **Table 1**. If for any reason it fails the date test, it is rescheduled to run in the next two (2) hours.

After successfully scheduling a job(s), the Hangfire Dashboard gives a graphical overview and status of each job on whether it was successfully executed or not, the scheduled jobs, the recurring jobs and much more. **Figure 4** shows the Hangfire Dashboard showing the recurring job.

API endpoints

API endpoints are the channels through which other applications can communicate or consume an API. **Table 2** lists the various endpoints that this API will provide. Endpoints are the point of entry, i.e., the Universal Resource Locator (URL). The state parameter is expecting a valid name of a state in Nigeria even states with more than one word such as *Akwa Ibom* are also acceptable. The date parameters take a date in the following format only DD-MM-YYYY.

Table 1. Cron expressions.

S/N	Field	Values	Meaning
1	min	0-59	For each minute
2	hour	0-23	Uses 24 hours format
3	day of month	1-31	1 for 1 st day of the month and increases till month-end
4	month	1-12	1 is for January and increases till December
5	day of week	0-6	0 is for Sunday and increase till Saturday

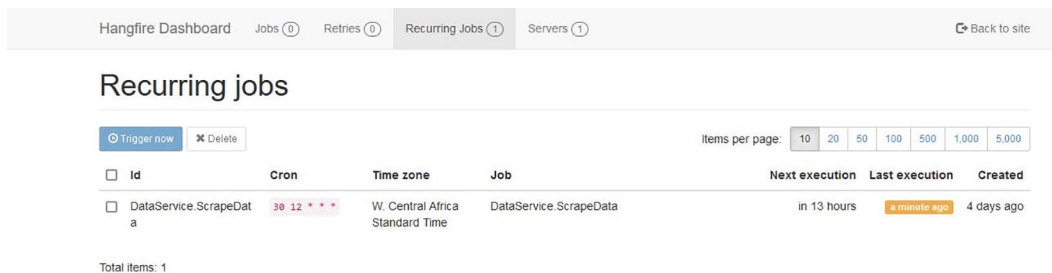


Figure 4. Hangfire dashboard showing recurring jobs.

Table 2. API endpoints.

Url/Endpoint	Verb	Operation	Description	Success	Failure
api/data/	GET	READ	Read all resources. Show only the latest data	200 OK	400 Bad Request 404 Not Found
api/data/s/{state}	GET	READ	Read all the data of a particular state	200 OK	400 Bad Request 404 Not Found
api/data/d/{date}	GET	READ	Read all the data of a particular date	200 OK	400 Bad Request 405 Not Allowed
api/data/{date} & {date}	GET	READ	Read all the data of a particular date range	200 OK	400 Bad Request 405 Not Allowed
api/data/{state} & {date} & {date}	GET	READ	Read all the data of a particular state on a certain date range	200 OK	400 Bad Request 405 Not Allowed

Implementation

The API can be access using any Web Browser or Software for consuming APIs such as Katalon Studio, Parasoft SOAtest, Eggplant, Postman etc. [Figure 7](#) shows a Graphical User Interface powered by Swagger that makes the API usable even by non – technical person while software such as Postman, Eggplant etc. need a level a technical understanding to know how to consume the API.

Operation

The API was designed with on a system with the following properties

- Processor: Intel(R) Core (TM) i5-4300U CPU @ 1.90 GHz 2.50 GHz
- RAM: 12GB
- OS: Windows 10 Pro x64
- IDE: Visual Studio 2019
- DBMS: Microsoft SQL Server Management Studio 18.9.2

A minimum software requirement needed to run this should be able to run Visual Studio 2019.²³

Results & discussion

This section briefly presents the results of the various tests conducted on the NigeriaCovid19API. Since this system is designed to fetch COVID19 data as showcased on the NCDC website and give a JSON response to API calls, data here is only warehoused and no manipulations were done on it. The tests were carried out to ascertain the functionality of the various API endpoints and ensure that data has been fetched successfully. This section presents the test we carried out with Postman, Swagger and MS SQL Server on a localhost computer.

Test with MS SQL Server

After data has been fetched, it was warehoused on an MS SQL Server and to confirm that the data has been stored successfully, we ran a READ operation (i.e., a SELECT instruction to fetch data from the database) to ascertain that data was successfully fetched and stored as shown in [Figure 5](#).

Test with Postman

Postman was developed to help with monitoring and testing APIs. We engaged it to inspect the API endpoints and their responses. [Figure 6](#) shows the JSON response for the */api/data* endpoint which gives the latest data in the database.

Test with Swagger

Swagger is a User Interface (UI) that gives a graphical interface for interacting with a REST API. It allows a user to visually interact and test an API. It uses the XML tags and annotations to get the output alongside the additional features for testing the API. [Figure 7](#) shows the general UI for Swagger and [Figure 8](#) shows the JSON response for */api/data*

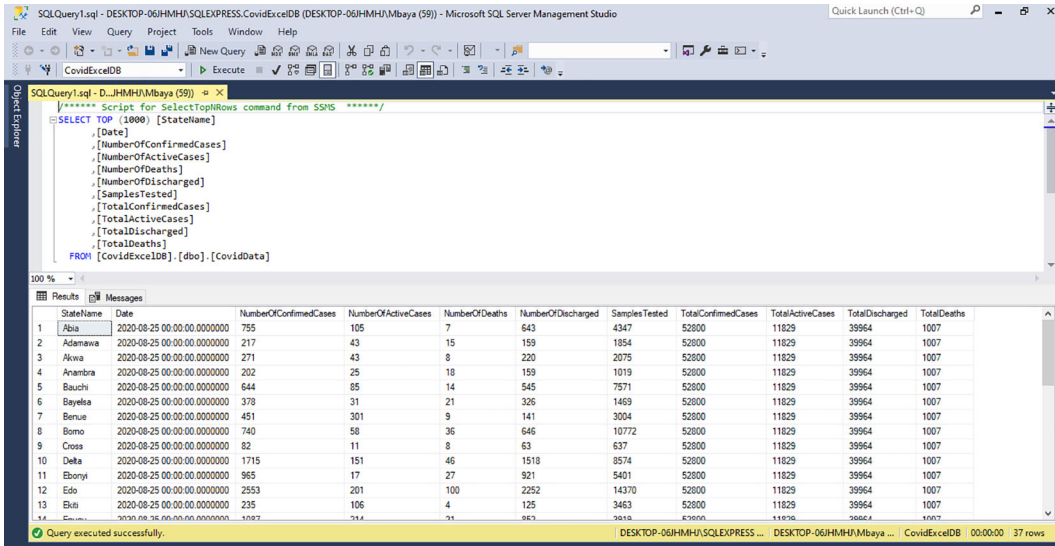


Figure 5. SQLQuery output.

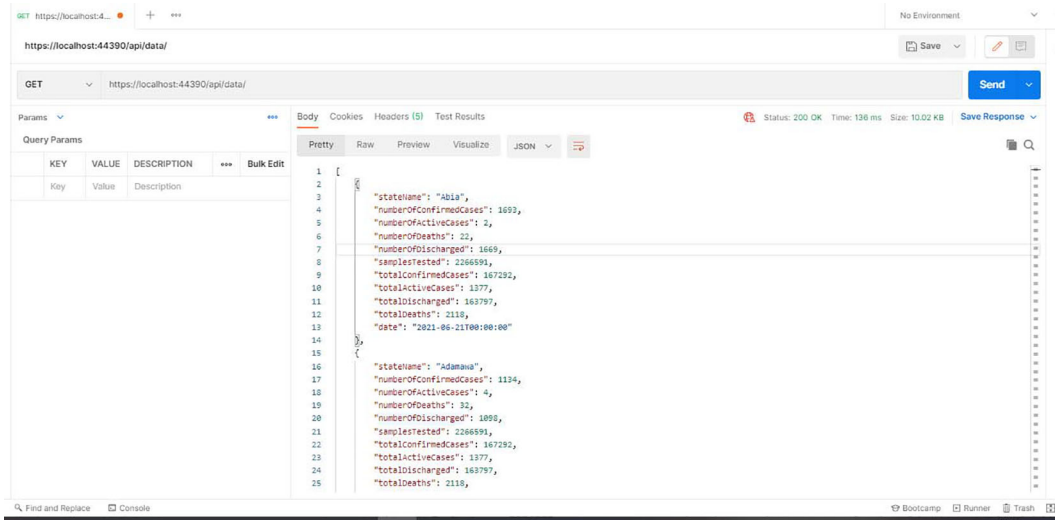


Figure 6. Postman JSON response for api/data.

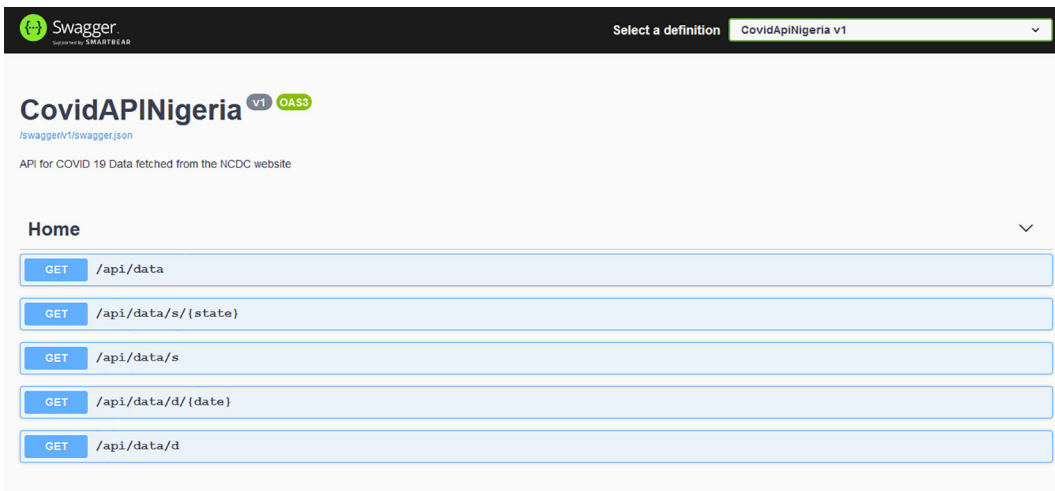


Figure 7. Swagger UI for the API.

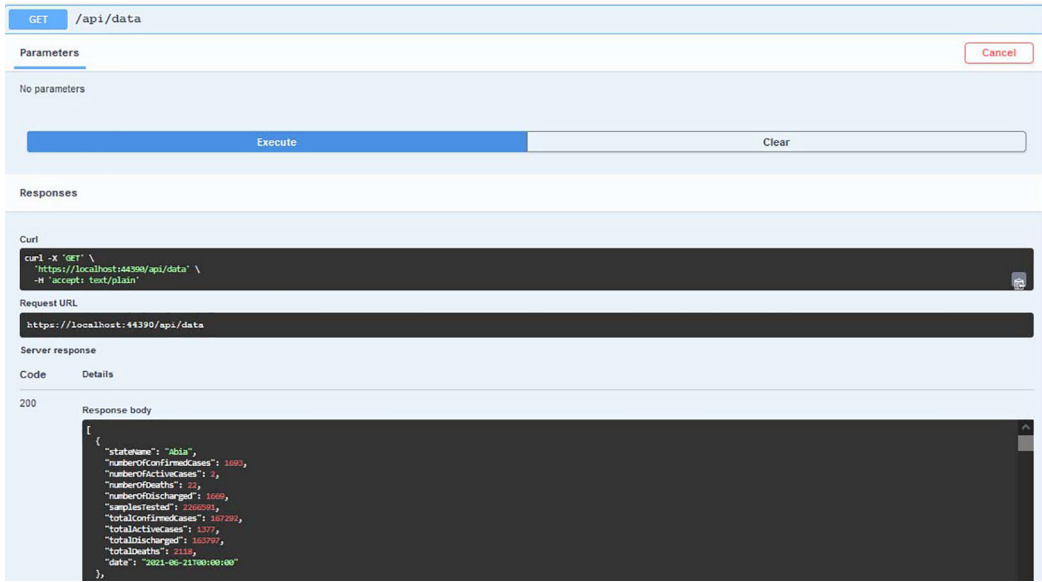


Figure 8. Swagger JSON response for `api/data`.

endpoint which gives the latest data in the database. Figure 9 shows the JSON response for `api/data/s/lagos`, which gives all the data for the Lagos state.

Testing with Blazer

BlazeMeter is an online open source-based, enterprise-ready platform that unifies all the functionality needed in carrying out various tests such as functional testing, performance testing, API testing and much more. API monitoring enables you

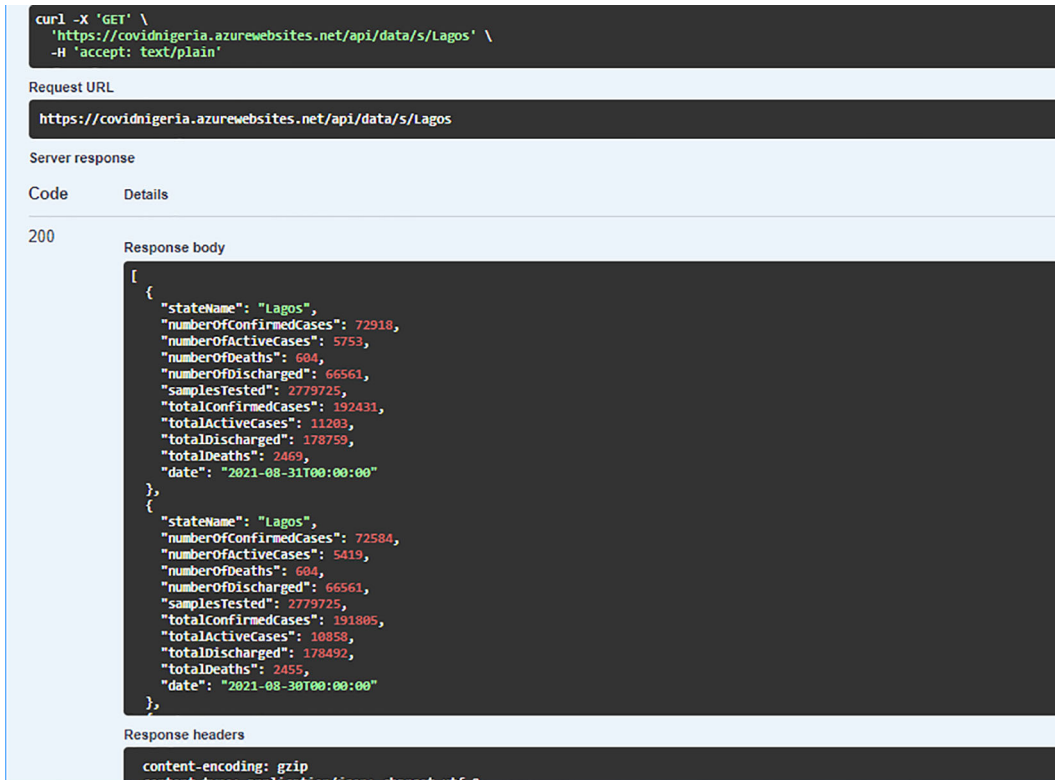


Figure 9. Swagger JSON response for `api/data/Lagos`.

to assess the impact of application performance enhancements. We may compare performance before and after the adjustments using historical response timing data.

To create a Test, logon to the BlazeMeter Website, Select Create Test, Select the Performance Test, Select the Enter URL and type in the URL request for the API, select the number of users to be used for testing, the duration (for the free package, you have to use the default location if you want to use a different location, you have to upgrade your account) and attach the necessary header variables were needed.

The API was tested on this platform and Figure 10 shows the summary, Figure 11 shows the response time and Figure 12 shows the hits. Throughput is the number of requests completed in a time interval and the system had an average of 11Hits/s with a response time of 2905 ms.

The volume of transactions created overtime during a test is referred to as throughput. It can also be described as the maximum capacity of a website or application. The number of concurrent users is the number of users who are actively using the app or website at any given moment, where we had 20 for the test. The number of hits per second generated by those concurrent users will be determined solely by their interactions with the app.

In Figure 11, Response Time is the time that passed to perform the request and receive a full response while the Latency is the time from sending the request, processing it on the server-side, to the time the client received the first byte. The time it

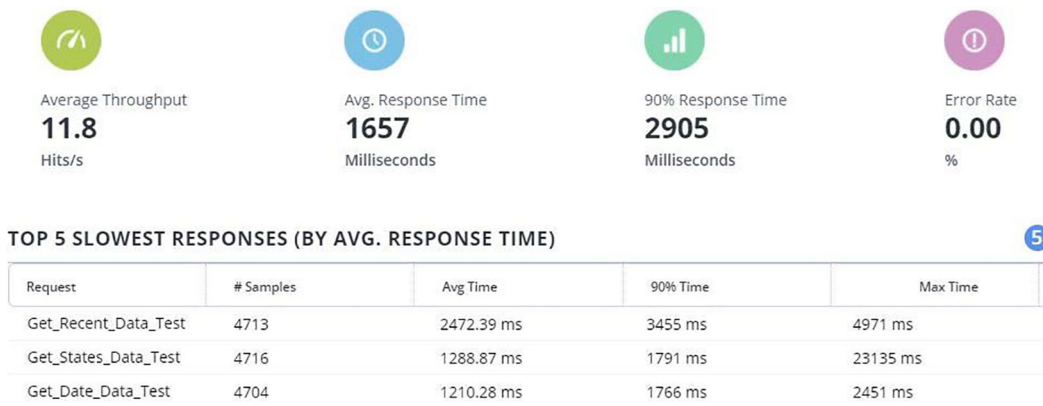


Figure 10. BlazeMeter result summary.

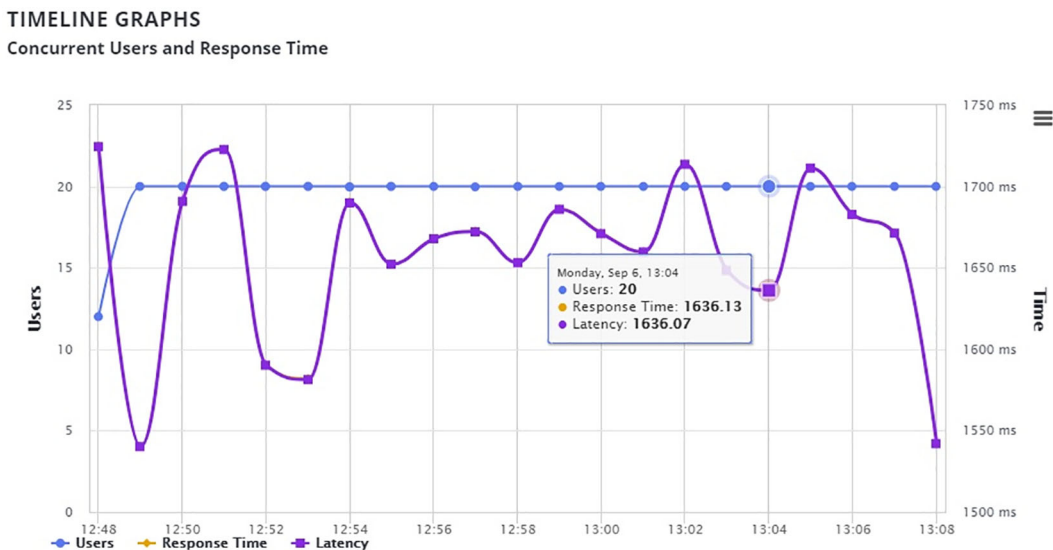


Figure 11. BlazeMeter response time results.

takes for data or a request to travel from the source to the destination is known as network latency. Network latency is measured in milliseconds.

In **Figure 12**, the number of HTTP requests sent by the user(s) to the Web server in a second is referred to as hits per second. The overall load put by concurrent virtual users on the server, regardless of whether they are executed successfully or not on the server-side, is measured in hits per second.

Plots from extracted data

Upon successful deployment²⁴ and testing of the API, data were extracted, and charts plotted to show the rate of infection, death and recovery rate and state with highest numbers as shown in **Figures 13, 14 and 15** respectively. **Figures 13 and 14** have two vertical lines, red and blue. The red line indicates the end of the first wave and commencement of the second wave while the blue signifies the end of the second wave and the start of the third wave.

TIMELINE GRAPHS

Concurrent Users and Hits/s

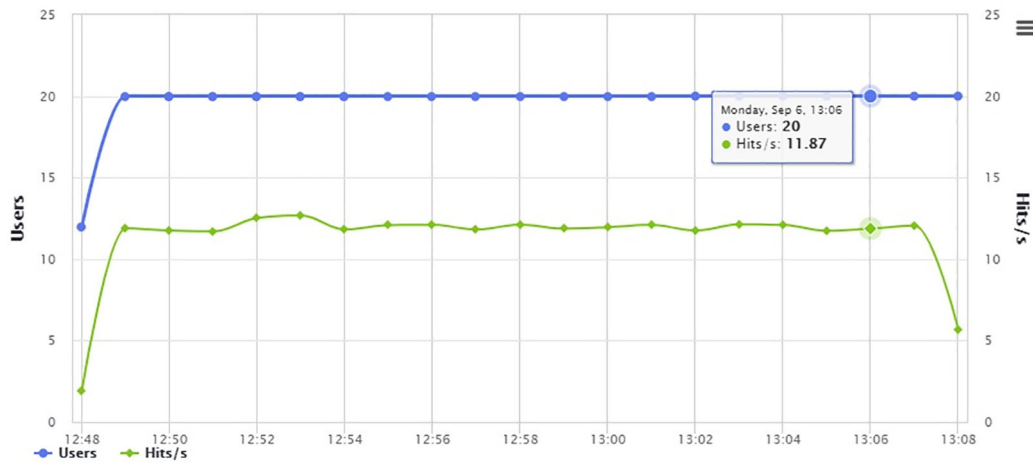


Figure 12. BlazeMeter hits per user result.

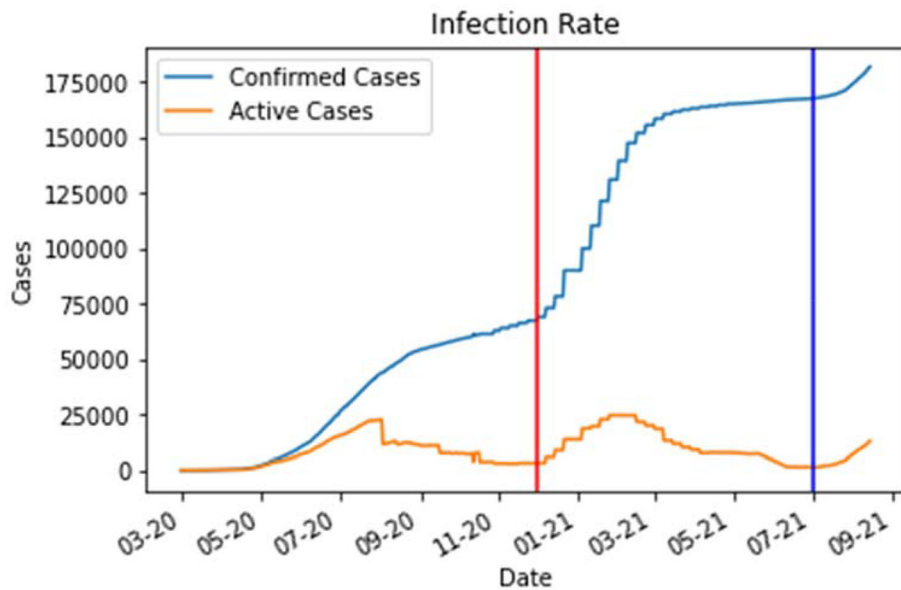


Figure 13. Covid-19 Nigeria - infection rate.

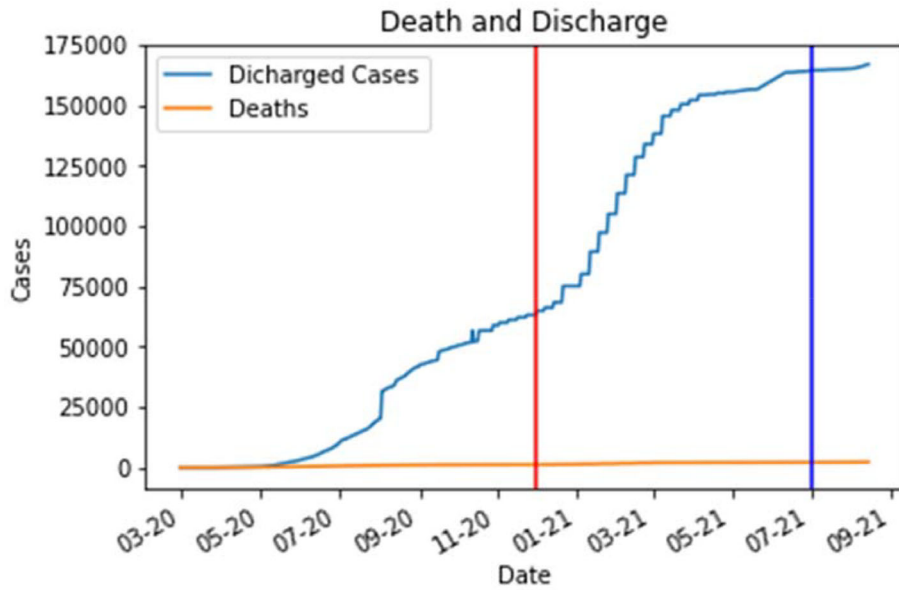


Figure 14. Covid-19 Nigeria – death and recoveries.

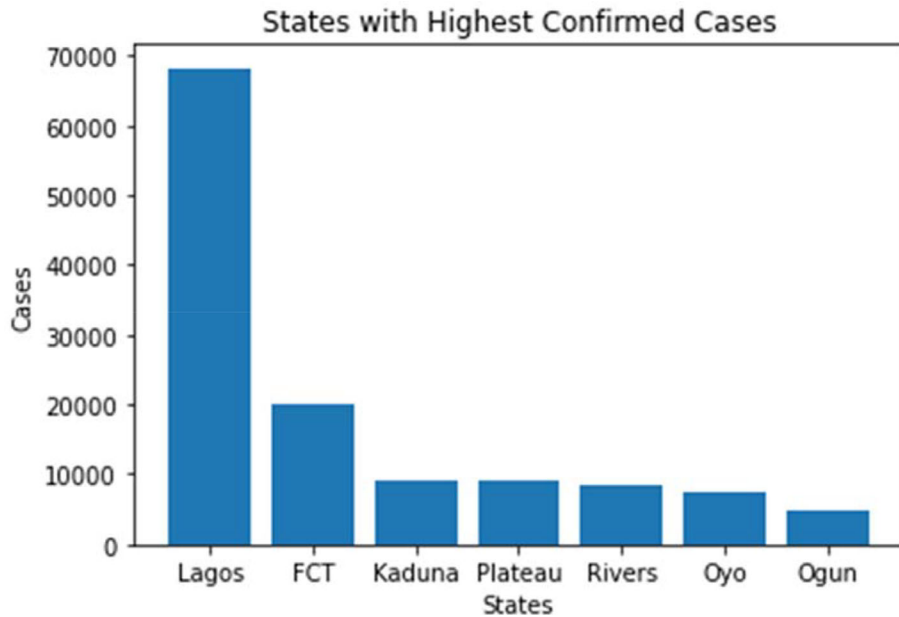


Figure 15. Covid-19 Nigeria – most affected states.

Initially as seen in Figure 13, the spread in Nigeria was slow with single digits after the first case was recorded, mainly at Lagos and Abuja but as the days goes by the number of active cases grew to reach a maximum of 12, 915 (as at August 2020) cases in Lagos state and the nationwide total confirmed cases kept raising to around 182,000 as at August 2021.

The first death in Nigeria as a result of COVID-19 was recorded on the 23rd of March 2020, the person had returned from medical treatment from the United Kingdom and with the rising number of cases, the Federal Government of Nigeria under the advice of the Presidential Task Force on the Control of COVID-19 placed a ban on all international flights in and out of the country except for emergency and essential flights. The death has continued to increase from Ogun state to other experiencing deaths of patients and hitting a peak of 465 deaths in Lagos state, however, many patients were able to recover and were discharged from the centres with a total of 166,826 persons across the nation as seen in Figure 14. The Line graph in the Figure looks like a straight line and that is because, in comparison with the Total Rate of Discharge, the total deaths seems small with a peak value of 2219 as of August 2021.

Lagos state is the hardest hit in terms of the number of confirmed cases with a total of 68220 cases, FCT Abuja reached 20158 cases, Kaduna 9195 cases, Plateau having 9205, Rivers having 8486 cases, Oyo with 7571 and Ogun having 5035 cases. Figure 15 shows the most affected states with a total confirmed case above 5,000 as of August 2021. The high level of cases in Abuja and Lagos can be attributed to the active community case search and testing. This public health response has led to the detection of more asymptomatic cases at the community level as stated by Ref. 2.

Nigeria shares land borders with the Republic of Benin in the west, Chad and Cameroon in the east, and Niger in the north. Figure 16 shows the confirmed cases in comparison with border countries and Nigeria having the highest number of confirmed cases and followed by Cameroon with cases above 75,000 cases as of August 2021 while Benin, Nigeria and Chad having cases below 25,000 cases.

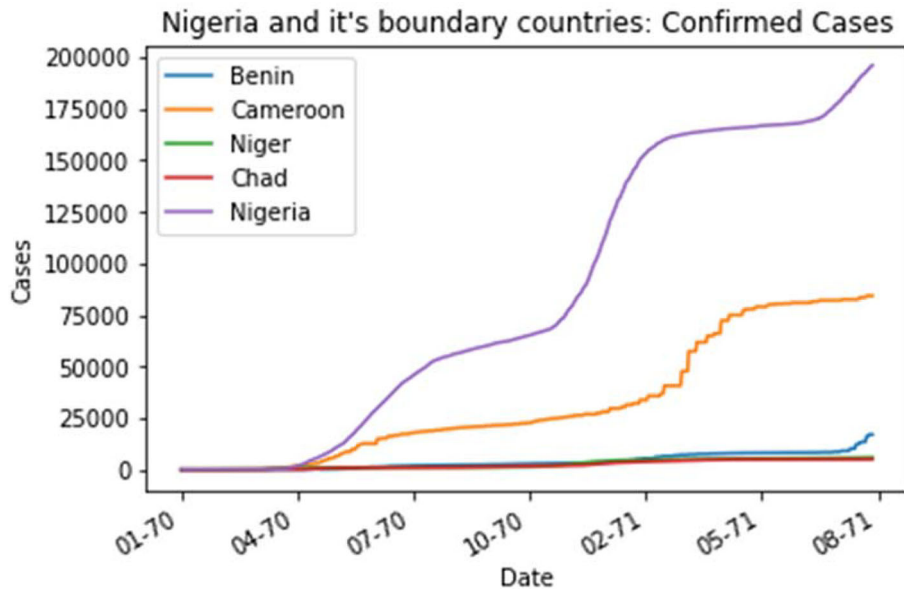


Figure 16. Nigeria and boulder countries COVID19 confirmed cases.

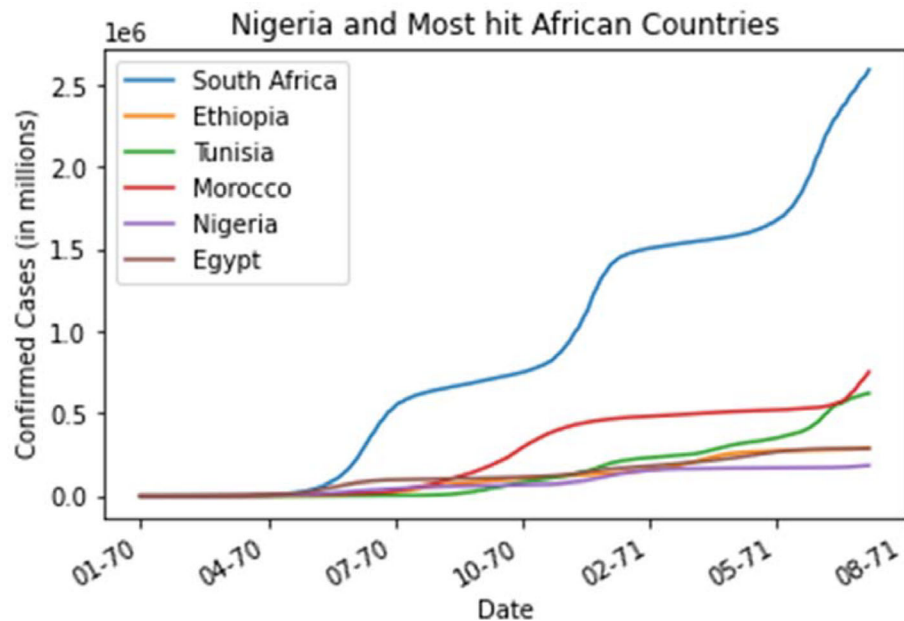


Figure 17. Nigeria vs top 5 most affected African countries.

As shown in [Figure 17](#), in comparison with the top (5) hit countries on the continent of Africa, Nigeria is the tenth on the continent as of August 2021 in terms of the number of confirmed cases with 180,000 cases while South Africa was the most affected on the continent with over 2.5 million cases, followed by Morocco over 750,000 cases, then Tunisia coming forth with about 600,000 cases and Ethiopia coming fifth on the continent with around 300,000 cases.

Conclusion

Knowing the importance of data to research, we have designed and developed an API that gives a JSON response to HTTP requests. It was built using the REST architecture and it is designed to fetch data from the NCDC website daily with the data being warehoused on MS SQL Server. Postman and Swagger were used to testing that Data endpoints to ensure the appropriate data is fetched as defined by the API endpoint, SQL Queries to select data were used to ensure that the database is properly populated with the right data and BlazeMeter was used to test the performance of the whole system and it achieved an 11 Hits/s with a response time of 2905 ms. Its advantage over existing COVID19 APIs for Nigeria is the access to previous data which means that researchers and data enthusiasts can use the API to data as required by their various research needs with a simple URL rather than having to comb through the archives of the NCDC website to get the data they need. Simply put, this system makes Nigeria's COVID19 data access to be easily accessible even by non-technical individuals with our user interface powered by swagger. The source codes and the dataset used in this paper are available as open-source on GitHub.^{25,26}

Data availability

Software availability

Software available from: <https://zenodo.org/record/5726851>²⁶

Source code available from: <https://github.com/emmaraj/CovidApiNigeria>²⁵

Archived source code as at time of publication: <https://doi.org/10.5281/zenodo.5726851>²⁶

License: MIT License

Acknowledgements

The authors acknowledge the financial support offered by Covenant University in the actualization of this research work for publication.

References

- Johnpaul M, Madububa BI: **The Trend of COVID-19 in Nigeria The Trend of COVID-19 in Nigeria**. 2020; vol. 16(4): pp. 43–51. [Publisher Full Text](#)
- Abdullahi IN, Emeribe AU, Ghamba PE, *et al.*: **Public health, socioeconomic responses, and associated challenges against coronavirus disease 2019 pandemic in nigeria**. *Open Access Maced. J. Med. Sci.* 2020; 8(T1): 122–125. [Publisher Full Text](#)
- Dg M, Yt G: **COVID-19 In Nigeria: Where are We?**. *Yenagoa Med. J.* January, 2021; 3(1).
- WHO Coronavirus (COVID-19) Dashboard | WHO Coronavirus (COVID-19) Dashboard With Vaccination Data**. (accessed Apr. 24, 2021). [Reference Source](#)
- Dr Osagie Ehanire: **Nigeria Centre for Disease Control**. 2020. (accessed Apr. 13, 2021). [Reference Source](#)
- NCDC Coronavirus COVID-19 Microsite**. (accessed Apr. 24, 2021). [Reference Source](#)
- Data | Definition of Data by Merriam-Webster**. (accessed Apr. 24, 2021). [Reference Source](#)
- Research | Definition of Research by Merriam-Webster**. (accessed Apr. 24, 2021). [Reference Source](#)
- Tripathi M, Chand M, Sonkar SK, *et al.*: **A brief assessment of researchers' perceptions towards research data in India**. *IFLA J.* 2017; 43(1): 22–39. [Publisher Full Text](#)
- Amira F, *et al.*: **CoronaTracker: World-wide COVID-19 Outbreak Data Analysis and Prediction**. March, 2020.
- McKinney W, *et al.*: **Pandas: A Foundational Python Library for Data Analysis and Statistics**. *Python High Perform. Sci. Comput.* 2011; vol. 14(no. 9): pp. 1–9.
- Ofoeda J, Boateng R, Effah J: **Application programming interface (API) research: A review of the past to inform the future**. *Int. J. Enterp. Inf. Syst.* 2019; 15(3): 76–95. [Publisher Full Text](#)
- Levine D: **APIs are the next big SaaS wave** | TechCrunch. Sep. 06, 2019. (accessed Apr. 26, 2021). [Reference Source](#)
- Kene-Okafor T: **Why API fintech startups are becoming a big deal in Nigeria – Techpoint Africa**. Nov. 04, 2020. (accessed Apr. 26, 2021). [Reference Source](#)
- covid-19-nigeria-api**. (accessed Apr. 26, 2021). [Reference Source](#)
- Corr AR, Federal U, Sobral C: **Application for predicting breast cancer through Google Prediction API**. vol. 458(no. July): pp. 50–53. 2018.
- Alag S: **Analysis of COVID-19 clinical trials: A data-driven, ontology-based, and natural language processing approach**. *PLoS One.* 2020; 15(9): e0239694–e0239613. [PubMed Abstract](#) | [Publisher Full Text](#)

18. Mackey TK, *et al.*: **Big data, natural language processing, and deep learning to detect and characterize illicit COVID-19 product sales: Infoveillance study on Twitter and Instagram.** *JMIR Public Heal. Surveill.* Jul. 2020; **6**(3).
[PubMed Abstract](#) | [Publisher Full Text](#)
19. Ulahannan JP, *et al.*: **A citizen science initiative for open data and visualization of COVID-19 outbreak in Kerala, India.** *J. Am. Med. Inform. Assoc.* Dec. 2020; **27**(12): 1913-1920.
[PubMed Abstract](#) | [Publisher Full Text](#)
20. Padmanaban R, Thirumaran M, Anitha P, *et al.*: **Computability evaluation of RESTful API using Primitive Recursive Function.** *J. King Saud Univ. - Comput. Inf. Sci.* 2018.
[Publisher Full Text](#)
21. Katayama T, Nakao M, Takagi T: **TogoWS: integrated SOAP and REST APIs for interoperable bioinformatics web services.** *Nucleic Acids Res.* Jul. 2010; **38**: W706-W711.
[PubMed Abstract](#) | [Publisher Full Text](#)
22. Fielding R: **Architectural Styles and the Design of Network-based Software Architectures.** 2000.
23. Microsoft: **Visual Studio 2019 System Requirements | Microsoft Docs.** (accessed Nov. 26, 2021).
[Reference Source](#)
24. **NaijaCovidAPI.** Aug. 30, 2021. (accessed Aug. 30, 2021).
[Reference Source](#)
25. Mbaya E: **CovidApiNigeria.** *Github.* Sep. 06, 2021. (accessed Sep. 06, 2021).
[Reference Source](#)
26. Baldwin ME: **emmaraj/CovidApiNigeria: NaijaCovidAPI (v1.0.0).** *Zenodo.* 2021.
[Publisher Full Text](#)

Open Peer Review

Current Peer Review Status:   

Version 1

Reviewer Report 20 June 2024

<https://doi.org/10.5256/f1000research.78809.r280858>

© 2024 Zunino A. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



Alejandro Zunino 

ISISTAN Research Institute, CONICET-UNICEN, Buenos Aires, Argentina

The paper describes the design, implementation and empirical evaluation of a REST API for accessing Nigeria's COVID19 data. The idea is valuable, since it enables researchers to consume data through the API for any purpose. The paper has two issues that needs to be addressed. First, its focus it not clear. It first describes the importance of APIs, REST and related technologies. This is aligned with the paper goals. However, the paper then focuses on discussing COVID data (Figures 12-17), which is out of the scope of an API and the paper. Second, assuming the core of the paper is the API and its design, from the description shown in Figure 3, it is clear that its design does not follow good practices such as hexagonal or clean architecture. Note that the data model is tied to the persistence and the API in HomeController. As a consequence, a minor change such as exposing the API using a different technology would require heavy refactoring of the code. The paper should be improved by focusing on the API and improving its design/implementation following good practices.

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Partly

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Yes

Are the conclusions about the tool and its performance adequately supported by the

findings presented in the article?

No

Competing Interests: No competing interests were disclosed.**Reviewer Expertise:** software engineering**I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.**

Reviewer Report 25 May 2024

<https://doi.org/10.5256/f1000research.78809.r226449>

© 2024 Olaolu Arowolo M. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**Micheal Olaolu Arowolo**

University of Missouri,, Columbia, USA

Review of literature should discuss limitations of existing studies

more review of literatures should be studied "Reference 1 "

Discuss figures, tables, algorithms used extensively

Discus dataset extensively

In results and discussion, discuss work done, results obtained and compare with existing works in literature

Conclusion should discuss work done, results obtained, significance, limitations, and future plans

References

1. Department of Computer Science, Kwara State University, Malete, Nigeria Department of Electrical Engineering and Computer Science, University of Missouri, Columbia, USA Department of Computer Science, University of Ilorin, Ilorin, Nigeria E-mails: *abdulrafiu.isiaka@kwasu.edu.ng, Isiaka R, Babatunde R, Ajao J, et al.: A Study of Impacts of Artificial Intelligence on COVID-19 Prediction, Diagnosis, Treatment, and Prognosis. *Advances in Multidisciplinary and scientific Research Journal Publication*. 2022; **10**: 11-62 [Publisher Full Text](#)
2. Jose AM, Samarpita S, Panchal NK, Sabina EP, et al.: Selective blockade of IL-21 by myricetin impedes T follicular helper cell differentiation by negatively regulating the JAK/STAT/Bcl-6 pathway in a rheumatoid arthritis animal model. *Biotech*. 2024; **14** (1): 25 [PubMed Abstract](#) | [Publisher Full Text](#)

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Bioinformatics, Machine learning, Data science, Computer science

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Reviewer Report 13 October 2023

<https://doi.org/10.5256/f1000research.78809.r209133>

© 2023 Li X. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



Xiaojin Li

Texas Institute for Restorative Neurotechnologies, The University of Texas Health Science Center at Houston, Houston, Texas, USA

This study presents the development of a COVID-19 API in Nigeria, built with REST architecture, ASP.NET Core Web API, and deployed on Microsoft Azure, enabling data retrieval from the NCDC website, with key findings highlighting the impact of COVID-19 in Lagos and Abuja, and its contribution as a comprehensive data hub for COVID-19 incidence in Nigeria from February 29, 2020, to the present. The following comments are intended to improve the quality of the paper.

1. This study lacks substantial innovation, as the author primarily demonstrates the creation of the COVID-19 dataset API using existing tools and tests it with commonly utilized API testing tools.
2. The API's functionality is somewhat limited, focusing primarily on supporting date and regional queries. The potential for more complex queries, though challenging, could significantly enhance its utility and contribution.

3. There is an absence of discussion regarding the scalability of the application. While the dataset of 200,000 people is relatively small in the context of COVID-19 studies, it would be interesting to explore the performance of the application with a dataset of several million people and its adaptability for use with other epidemiological electronic health record (EHR) data.

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

No

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

No

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Ontology-driven Clinical Data Management including data capture, integration, cross-cohort faceted query, data visualization, and temporal query.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to state that I do not consider it to be of an acceptable scientific standard, for reasons outlined above.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research