

Removing Embargo of PHP from Communicating With Microsoft SQL Server Database: Functional Discrepancy between Microsoft Server Database and MYSQL Database

Onyeacholem Ifeanyi Joshua*
Software Developer and Data Scientist
Delta State University
Abraka, Nigeria.

Abstract:- Over the years the role of data when it comes to analysis, interpretation, and prediction can't be overemphasized, the effectiveness and the interaction of any application hinged on inputting and outputting data to ascertain its functionalities and robustness. We are looking at how we can create data and get it inserted into two different databases same time, with the same data type without any constraint, this is very important where we have a cloud (server database) that only takes input from a particular technology, say (.Net Framework) and we have another solution running on PHP technology. In this paper, we were able to implement different applications technology to exchange data from two different databases so as to communicate seamlessly with each other through API (Application Program Interface). The need to fetch data between two applications is to enable two applications to communicate seamlessly without interference and alteration from one application database to the other. The role of API is like giving a visitor your house key but the visitor can only enter the room that the key is meant for (not the whole room), for the cause of this paper we used MySQL Server database and MySQLi database (Maria DB) and the result on Scalability and Performance of these two databases synchronization was accurate.

Keywords:- Authentication; SQL; Server; Database; API; curl; Microsoft.

I. INTRODUCTION

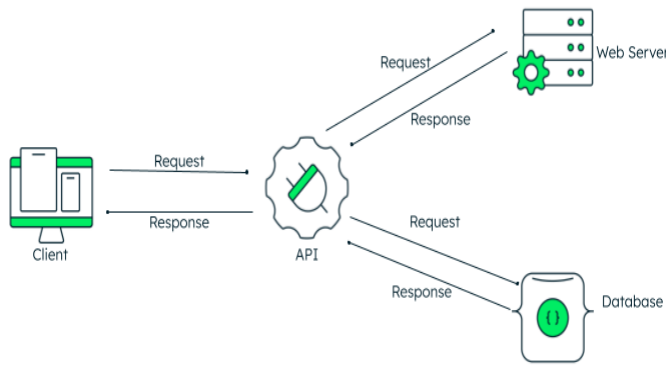
Glassdoor 2020; affirmed that each person generates 1.7 megabytes of data every second. All the data generated by these individuals are stored in databases and can be analyzed for better decision-making, that is to say in the absence of data, analysis, prediction, and prescription is in question (not feasible).

Spencer (2022); confirmed that Gartner predicts that global government IT spending will reach nearly \$558 billion in 2022, a 6.5% increase over the previous year. As technology integration becomes more common in society, the government, organizations, and companies must begin the digital transformation process. APIs and databases are at the

heart of effective digital transformations. A database is a structured collection of information that is kept in a way that allows users to easily access, manage, and change it. In general, organizations use databases to store, maintain, and retrieve any type of data. A database's main function is to enable organizations to make more informed business decisions based on well-organized data and information.

Tom (2022) stated that a microservice framework is a type of middleware that creates a consistent foundation for running microservices. The framework's purpose is to make common element implementation easier and more consistent. This is crucial for accelerating microservices development and deployment without imposing a significant operational burden. Microservice frameworks are usually tailored to a single application paradigm, such as web front-end or event processing. Frameworks divide into two categories: proprietary, like Microsoft's .NET, and cross-platform, such as Java. The most popular frameworks for hybrid cloud front-end APIs are Drop-wizard, Micronaut, Molecular, Rest-let, Spark, Spring Boot, and Vert. x.

An API, or Application Programming Interface, acts as a courier or intermediary between computer programs, allowing them to safely obtain data from one another. A ticket scheduling website that analyses costs from various carriers is a typical example of an API. The program obtains data from a number of third-party APIs, which then link to the real data source. (Like databases). The client user interface (UI) shows the information in a single view after the API sends the answer to the asking application. A database or web server is not an API. However, it offers both of them safe entry. When a client or program asks the API for data, the API routes the request to the proper source and sends the obtained answer to the client.



What API is and what does it do

Fig 1: <https://www.mongodb.com/databases/what-is-an-api>

Assume we want to create an external app that allows users to access specific information in an application database, say, student details are stored in this specific table saved in the database, for an external application to get access to these data, the internal application (the original owner of the database) will create an API, configure it and structure it to the required information needed from the external application with some security constraints to grant it access to the intended users.

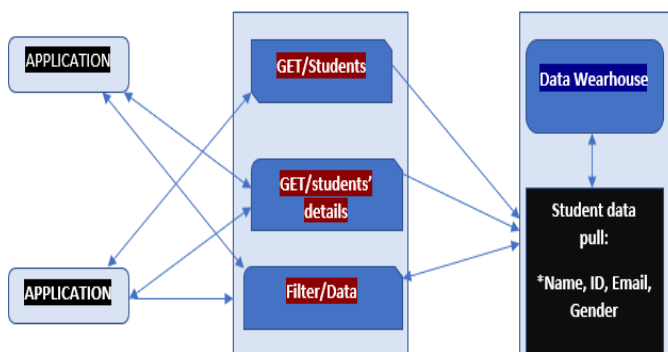


Fig 2. Showing the function display on fetching and filtering specific students' data

We routinely extract data from web APIs to show to end users while constructing a client-side online application, and if end users modify/upload existing/new data, we must verify that the CRUD action is called effectively. Several current systems interact with servers using the client-server paradigm.

Ahmed (2016) demonstrated how you might ask a shopkeeper to take a flower (or a bundle of flowers), put it in a wrapper (and possibly brand it), and offer it to you. Just the merchant has access to the bunches, and he begins his duty after you pay. And only he knows where and how the bloom is kept. to get there. He will wrap it himself and label it with your name. And he should give it to you on occasion or mail it to an address you have provided. Likewise, the software requests data from an API while passing the filter conditions and arguments. To connect to a database, API access credentials are used (read hits the database).

Anuff (2022), an application has limited value until it is put into production. Getting to this point as rapidly as possible means that developers will have easy access to the data, they need to build without having to worry about the difficulties of spinning up, managing, and maintaining databases. Although this was not always the case, APIs are now the de facto standard for connecting apps to databases. In this section, we'll look at what's transpired in the software world to make exposing databases as APIs increasingly significant. We'll also discuss Stargate, an open-source solution that helps developers cope with these APIs. The most recent version of Stargate adds scalability and versatility.

Santa (2022) stated that DataStax, the real-time data firm, announced the introduction of Stargate v2, a new version of the award-winning open-source data gateway that allows application developers to build real-time apps for Apache Cassandra® using their preferred API, according to Santa (2022). Stargate v2, an open-source database-as-a-service from DataStax, now offers a high-performance Grpc API, allowing users to effortlessly expand Cassandra data to serve billions of global devices in real time. "We've relied on DataStax's solutions - Astra DB's managed Cassandra database and Stargate Grpc API - to ensure our customers have quick access to data and insights," said Deepak Kumar, SHIELD's VP of Engineering.

Perry (2017) noted that When individuals use the term "API," they often generalize and mean "a publicly available web-based API that returns data, most likely in JSON or XML." The API is not the database or even the server; it is the code that manages the server's access point(s). You could email Twitter and request a spreadsheet containing all of these tweets. But then you'd have to figure out how to integrate that spreadsheet into your program, and even if you stored them in a database, as we did, the data would quickly become outdated. It would be impossible to keep up with the changes. It would be preferable and easier for Twitter to give you the means to query their application to obtain that information.

Okpalla (2023) In today's knowledge-based society, where computers are present in every aspect of society, database systems perform crucial functions. As a result, both database users and developers have expressed great concern about how databases have changed over time. The necessity to follow the trend also arises when data volume and speed continue to rise.

II. METHODS SPECIFICATION

In the development process, we employed the following methods: NotePad++ as our IDE (integrated development environment) as our code editor, with two distinct connection strings to two different databases, and PHP as our programming language of choice. We created a database name DemoDB and a table inside it named users. By default, PHP can't use the Microsoft SQL Server database so we had to do some configuration in our PHP (php.ini file) by adding two more extensions to the extension folder, this is because

many DLL files are located in the extensions folders. The following two lines are the extensions:

- extension=php_pdo_sqlsrv_74_ts_x64.dll
- extension=php_sqlsrv_74_ts_x64.dll

The first one is to enable a smooth handshake between the Microsoft SQL server database and PHP to interchange data and communicate seamlessly using PDO (PHP Data Objects) while the second extension will enable us to use some of the Microsoft SQL server built-in functions in PHP to write our queries effectively.

```

1 <?php
2 $serverName = "DESKTOP-2QUS0SK";
3 $connectionOptions = [
4     "Database" => "DemoDB",
5     "Uid" => "",
6     "PWD" => ""
7 ];
8
9 $conn = sqlsrv_connect($serverName,$connectionOptions);
10 if($conn == false) die (print_r(sqlsrv_errors(),true));
11 //else echo 'Connection successfull';
12
13 ?>
    
```

Fig 3: Showing connection string query for Microsoft SQL Server database.

```

1 <?php
2
3 $link = mysqli_connect("localhost","root","","demodb") or die ("Unable to connect");
4
5
6 ?>
    
```

Fig 4: Showing connection string query for MySQLi database (MariaDB).

III. DISCUSSION AND RESULTS

Register Student Details

Student ID: User123

Title: Mr. (dropdown)

Surname: Onyeacholem

Other Names: Joshua Ifeanyi

Sex: Male (selected), Female

Marital status: Single (dropdown)

Phone Nnumber: 08065800282

Email ID: brandykoke@gmail.com

Student password: josh123 (with eye icon)

Submit

Fig 5: Showing registration form for student details.

The student registration form takes data as inputs and saves it into multiple databases, from the above snapshot we can see that the form has been filled, and once it is submitted, it inserts data into the two different databases.

+ Options

id	staffid	titles	onames	sname	sex	mstatus	pno	email	pwd
18	User123	Mr	Joshua Ifeanyi	Onyeacholem	male	Single	08065800282	brandykoke@gmail.com	josh123

Options: Edit, Copy, Delete, Export

Fig 6: Showing MySQLi inserted result

staffid	titles	onames	sname	sex	mstatus	pno	email	pwd	
1	User123	Mr	Joshua Ifeanyi	Onyeacholem	male	Single	08065800282	brandykoke@gmail.com	josh123

Fig 7: Showing Microsoft SQL Server database result.

We can see that the data was inserted successfully into the MySQLi database (Maria DB) and also the Microsoft SQL Server database.

IV. API FETCH QUERY USING CURL

```

<<?php
if(isset($_POST["token"])){
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, 'https://studentportal.edu.ng/api/sso/validate?token='.$_POST["mat_no"]);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
    $headers = array();
    $headers[] = "Authorization:rSHdcSbCqaMq5trpiGq268zKvCcF4FloblewHADADFdsadFGddsadgdSDGvFDSads";
    $headers[] = "Accept: application/json";
    $headers[] = "Content-Type: application/json; charset=UTF-8";
    $headers[] = "APIKey: DLS-LMSM$SRPI";
    curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);

    $result = curl_exec($ch);
    curl_close($ch);

    $res=json_decode($result, true);
    $matricNumber=$res["data"]["biodata"]["matricNumber"];
    $surname=$res["data"]["biodata"]["surname"];
    $DepartmentName=$res["data"]["biodata"]["DepartmentName"];
    $FacultyName=$res["data"]["biodata"]["FacultyName"];
    $otherNames=$res["data"]["biodata"]["otherNames"];
    $dateOfBirth=$res["data"]["biodata"]["dateOfBirth"];
    $phoneNumber=$res["data"]["biodata"]["matricNumber"];
}
}

```

Fig:8 snapshot of API code Snippet.

With the above API code snippet, we can fetch data from any database be it Mongo, PostgreSQL, Microsoft SQL server, etc. but this research is limited to Microsoft SQL Server database.

RECOMMENDATION

From the above-proved analysis and implementation, it is, therefore, necessary to choose any of these databases for development purposes, irrespective of the type of technology(programming language) that is been used, it all depends on the developer and their level of programming technicalities in their styles of query structures.

REFERENCES

- [1]. Ahamed Yashif: How does an API work with a database(<https://www.quora.com/what-is-the-difference-between-API-and-database>)
- [2]. Ed Anuff: Why Database Need APIs(Oct 26 th , 2022, <https://www.thenewstack.ai/why-databases-need-apis>)
- [3]. Glassdoor 2020: Each person generates 1.7 megabytes of data every second (<https://www.glassdoor.com/index.html>)
- [4]. Perry Eising: What exactly is an API? And how an API works(Dec 7, 2017)
- [5]. SANTA CLARA, Caliuf:-Oct.26 th, 2022:DataStax Delivers Stargate v2:Unlocking Apache Cassandra Data to Ser Billions of Devices in Real Time.
- [6]. Spencer Nguyen: Understanding the connection between databases and APIs(2022).
- [7]. Chidimma Lilian Okpalla: Current computing trend in database design