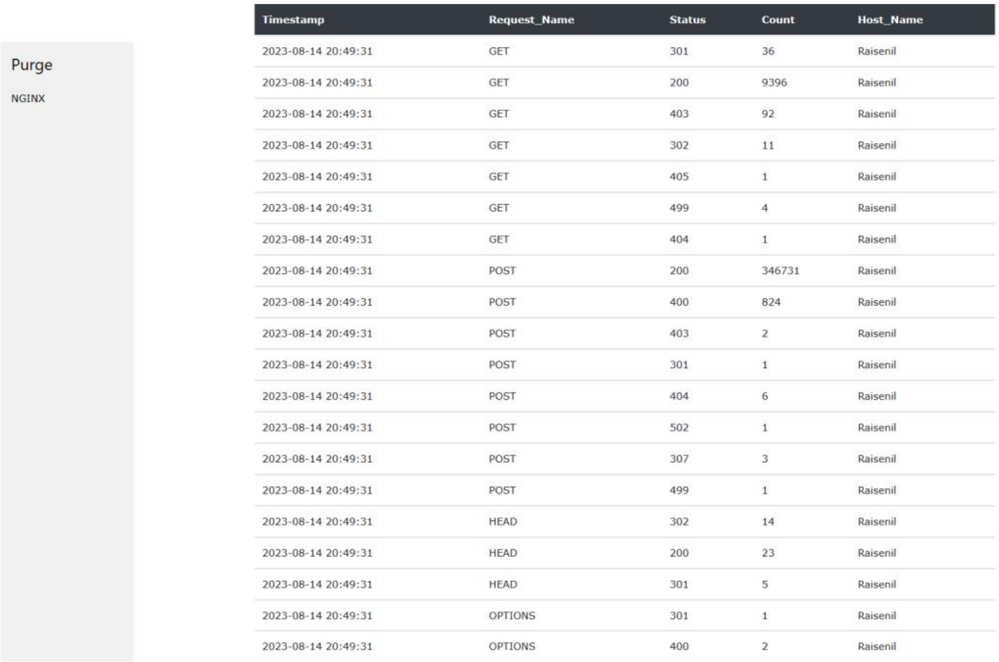
LoG Parser Service Documentation

Table of Contents

1. Introduction

Welcome to the documentation for the LoG Parser service Website. This service is designed to parse NGINX access logs, categorize requests by their HTTP methods (POST, GET, OPTIONS, HEAD, PUT, DELETE, PATCH), generate a CSV file, and store the parsed data in a database. The service is developed using PHP.



Timestamp	Request_Name	Status	Count	Host_Name
2023-08-14 20:49:31	GET	301	36	Raisenil
2023-08-14 20:49:31	GET	200	9396	Raisenil
2023-08-14 20:49:31	GET	403	92	Raisenil
2023-08-14 20:49:31	GET	302	11	Raisenil
2023-08-14 20:49:31	GET	405	1	Raisenil
2023-08-14 20:49:31	GET	499	4	Raisenil
2023-08-14 20:49:31	GET	404	1	Raisenil
2023-08-14 20:49:31	POST	200	346731	Raisenil
2023-08-14 20:49:31	POST	400	824	Raisenil
2023-08-14 20:49:31	POST	403	2	Raisenil
2023-08-14 20:49:31	POST	301	1	Raisenil
2023-08-14 20:49:31	POST	404	6	Raisenil
2023-08-14 20:49:31	POST	502	1	Raisenil
2023-08-14 20:49:31	POST	307	3	Raisenil
2023-08-14 20:49:31	POST	499	1	Raisenil
2023-08-14 20:49:31	HEAD	302	14	Raisenil
2023-08-14 20:49:31	HEAD	200	23	Raisenil
2023-08-14 20:49:31	HEAD	301	5	Raisenil
2023-08-14 20:49:31	OPTIONS	301	1	Raisenil
2023-08-14 20:49:31	OPTIONS	400	2	Raisenil

2. Technology Stack

The Inventory Management System is built using the following technologies:

PHP version : PHP 8.2.4 (cli) (built: Mar 14 2023 17:54:25) (ZTS Visual C++ 2019 x64)

Database : MySQL database system version - 10.4.28-MariaDB.

Frontend : HTML, CSS, Bootstrap 5.

Other : NGINX

3. Installation

To install the LoG Parser service, follow these steps:

1. Clone the repository from the GitHub repository: GitHub Repository
2. Navigate to the project directory:

```
cd LoG_Parser
```

4. Configuration

The configuration for the LoG Parser service is stored in the `Model/db.php` file. Open this file and update the following configuration options:

- `$HOST`: The hostname of the database server.
- `$DB_USERNAME`: The database username.
- `$DB_PASSWORD`: The database user's password.
- `$DB_NAME`: The name of the database.
- `$filename`: The path to the NGINX access log file.

5. Usage

To use the LoG Parser service, follow these steps:

1. Ensure that the NGINX access log is being written to the specified `$filename`.
2. Download and install XAMPP from the official website (`https://www.apachefriends.org/index.html`). Follow the installation instructions for your operating system.
3. Make sure XAMPP is up and running. Start the Apache and MySQL modules from the XAMPP control panel.

4. Create a New Database by going On the `phpMyAdmin` interface, click on “New” in the left sidebar to create a new database. Enter name as `log_parser` and choose the appropriate collation (usually utf8_general_ci).

5. Import the Backup by selecting it from the left sidebar. Click on the “Import” tab in the top menu. Click the “Choose File” button to select the .sql backup file from `Database\log_parser.sql`.Choose the default settings or adjust them based on your needs.Click the “Go” button to start the import process.

6. Access the website in your browser at `http://localhost:3000`.

7. The website will process the access log, categorize requests by HTTP methods, generate a CSV file, and store the data in the database.

6. Database

The LoG Parser service uses a database to store parsed data. The data is stored in a table named `log`. The table schema includes columns for `id, Timestamp, Request_Name, count, and Host_Name`.

7. Regular Expressions

The service uses regular expressions to extract relevant information from the access log entries. Regular expressions are defined in the `Purging_Script/log.php` file. If your NGINX access log format differs from the default, you may need to modify these regular expressions to match your log format.

8. CSV Generation

The parsed data is used to generate a CSV file named `Purging_Script/CSV/log.csv`. The CSV file is created in the root directory of the project. The generated CSV file includes columns for `Timestamp_csv, Method, Status_Code, Count & Host_Name`.

9. Maintenance and Troubleshooting

- Log File Path: Ensure that the `$filename` in the `Model/db.php` file is correctly set to the NGINX access log file’s path.

- Regular Expressions: If your NGINX access log format changes, update the regular expressions in the `Purging_Script/log.php` file to match the new format.

- Database Connectivity: If you encounter database connectivity issues, verify the correctness of the database configuration in the `Model/db.php` file.

10. Conclusion

Congratulations! You’ve successfully set up and configured the LoG Parser. This service will parse the NGINX access log, categorize requests by HTTP methods, generate a CSV file, and store the parsed data in a database. If you have any questions or encounter issues, refer to this documentation or reach out to me.

For further updates and enhancements, please refer to the GitHub repository: GitHub Repository