SIT323/SIT737- Cloud Native Application Development

Name: Sizhe Wang

Student ID: 223314413

4.2C: Enhanced Functionality for the Calculator Microservice

GitHub Link: https://github.com/AndyWanng/sit323-737-2024-t1-prac4c.git

Enhancements Overview

The enhancements to the Calculator Microservice can be summarized in the following key areas:

1. **Extended Mathematical Operations:**
    - **Exponentiation (exponent):** Allows users to calculate the power of a number.
    - **Square Root (sqrt):** Provides the square root of a number. It includes error handling for negative inputs.
    - **Modulo (modulo):** Returns the remainder of the division of two numbers.
    - **Absolute Value (abs):** Returns the absolute value of a number.
    - **Remainder (remainder):** Similar to modulo.
2. **Improved Error Handling:**
    - Detailed error messages are now provided for specific error scenarios, such as attempting to divide by zero or finding the square root of a negative number.
    - General error handling has been enhanced to catch and respond to unexpected errors gracefully, providing a status code and a clear message.
3. **Enhanced Logging:**
    - The logging configuration has been upgraded to include timestamps and uppercase log levels for better readability and debugging.

- Error logs now provide more context, including the specific operation and input values that led to the error, facilitating easier diagnosis and resolution of issues.

Implementation Details

1. **Arithmetic Operations:** The service defines a set of functions for each supported operation, handling edge cases (e.g., division by zero) as necessary.
2. **Input Validation:** A middleware function validateInput ensures that all inputs are valid numbers and specific operations receive the correct number of inputs. It also logs invalid input attempts.
3. **Error Handling Middleware:** An Express middleware catches and handles errors thrown during operation processing, logging them and returning appropriate responses to the client.
4. **Logging Configuration:** Uses Winston to configure logging with timestamped entries and includes separate logs for errors and general information.

Getting Started

Prerequisites

To run this project, you'll need:

- Node.js installed on your machine. Download and install it from Node.js official website.
- NPM, which comes with Node.js.

Installation

1. **Clone the repository** or download the project files to a local directory.
2. **Navigate to the project directory** in your terminal or command prompt.
3. **Install dependencies** by running the command:

   npm install express winston

   This will install Express for the web server functionality and Winston for logging.

Running the Service

1. **Start the server** by running:

   node calculator.js

   If everything is set up correctly, you should see a message indicating that the server is running on port 3000.

2. **Accessing the service**: The service can now be accessed through HTTP GET requests. You can use a web browser, Postman, or curl to make requests.

API Endpoints

The service exposes four endpoints, one for each arithmetic operation. The endpoints are accessed via GET requests and expect two query parameters: num1 and num2, representing the operands.

- **Addition**
  - Path: **/add**
  - Example: **http://localhost:3000/add?num1=10&num2=5**
- **Subtraction**
  - Path: **/subtract**
  - Example: **http://localhost:3000/subtract?num1=10&num2=5**
- **Multiplication**
  - Path: **/multiply**
  - Example: **http://localhost:3000/multiply?num1=10&num2=5**
- **Division**
  - Path: **/divide**
  - Example: **http://localhost:3000/divide?num1=10&num2=5**
  - Note: Dividing by zero will return an error.
- **Exponentiation**
  - Path: **/ exponent**
  - Example: **http://localhost:3000/ exponent?num1=2&num2=3**
- **Square Root**

    - Path: **/ sqrt**
    - Example: **http://localhost:3000 /sqrt?num1=16**
- **Modulo**
    - Path: **/ modulo**
    - Example: **http://localhost:3000/ modulo?num1=10&num2=3**
- **Absolute Value**
    - Path: **/ abs**
    - Example: **http://localhost:3000/ abs?num1=-5**
- **remainder**
    - Path: **/ remainder**
    - Example: **http://localhost:3000/ remainder?num1=10&num2=4**

Error Handling

The service provides meaningful error messages for invalid input (e.g., non-numeric values) or unsupported operations.

Logging

Logs are generated using Winston and are divided into console logs for info level and file logs for error and combined logs. Logs are stored in the logs directory.

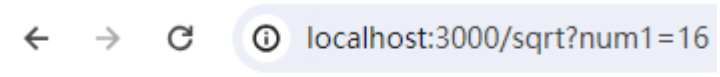
Conclusion

These enhancements significantly improve the Calculator Microservice's functionality, user experience, and maintainability. By documenting these changes, we ensure that users and developers can easily understand and utilize the service, while also facilitating future enhancements.

Screenshots:

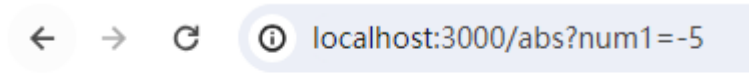
Extended Mathematical Operations:

{"statuscode":200,"data":8}

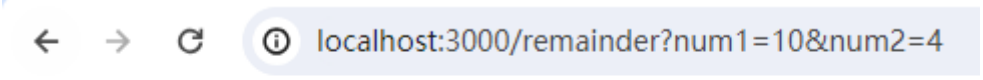


{"statuscode":200,"data":4}

{"statuscode":200,"data":1}



{"statuscode":200,"data":5}



{"statuscode":200,"data":2}

Enhanced error handling

```
PS C:\Users\22396\WebstormProjects\task-4.2c> node calculator.js
2024-03-14 11:36:22 INFO: Server running on port 3000
2024-03-14 11:37:15 INFO: Operation exponent successful on num1=2, num2=3, Result=8
2024-03-14 11:37:15 ERROR: Invalid input: num1=undefined, num2=undefined, operation=favicon.ico
2024-03-14 11:37:40 INFO: Operation sqrt successful on num1=16, num2=undefined, Result=4
2024-03-14 11:37:40 ERROR: Invalid input: num1=undefined, num2=undefined, operation=favicon.ico
2024-03-14 11:38:03 INFO: Operation modulo successful on num1=10, num2=3, Result=1
2024-03-14 11:38:03 ERROR: Invalid input: num1=undefined, num2=undefined, operation=favicon.ico
2024-03-14 11:38:28 INFO: Operation abs successful on num1=-5, num2=undefined, Result=5
2024-03-14 11:38:28 ERROR: Invalid input: num1=undefined, num2=undefined, operation=favicon.ico
2024-03-14 11:38:56 INFO: Operation remainder successful on num1=10, num2=4, Result=2
2024-03-14 11:38:56 ERROR: Invalid input: num1=undefined, num2=undefined, operation=favicon.ico
2024-03-14 11:39:20 ERROR: Operation not found: notAnOperation
2024-03-14 11:39:20 ERROR: Invalid input: num1=undefined, num2=undefined, operation=favicon.ico
2024-03-14 11:39:44 ERROR: Error occurred: Cannot find the square root of a negative number.
2024-03-14 11:39:44 ERROR: Invalid input: num1=undefined, num2=undefined, operation=favicon.ico
2024-03-14 11:40:04 ERROR: Error occurred: Cannot divide by zero.
2024-03-14 11:40:04 ERROR: Invalid input: num1=undefined, num2=undefined, operation=favicon.ico
```

Enhanced logs

```
2024-03-14 11:04:48 ERROR: Invalid input: num1=undefined, num2=undefined, operation=favicon.ico
2024-03-14 11:05:12 ERROR: Invalid input: num1=undefined, num2=undefined, operation=favicon.ico
2024-03-14 11:05:27 ERROR: Invalid input: num1=undefined, num2=undefined, operation=favicon.ico
2024-03-14 11:05:38 ERROR: Error occurred: Cannot find the square root of a negative number.
2024-03-14 11:05:38 ERROR: Invalid input: num1=undefined, num2=undefined, operation=favicon.ico
2024-03-14 11:05:59 ERROR: Invalid input: num1=undefined, num2=undefined, operation=favicon.ico
2024-03-14 11:37:15 ERROR: Invalid input: num1=undefined, num2=undefined, operation=favicon.ico
2024-03-14 11:37:40 ERROR: Invalid input: num1=undefined, num2=undefined, operation=favicon.ico
2024-03-14 11:38:03 ERROR: Invalid input: num1=undefined, num2=undefined, operation=favicon.ico
2024-03-14 11:38:28 ERROR: Invalid input: num1=undefined, num2=undefined, operation=favicon.ico
2024-03-14 11:38:56 ERROR: Invalid input: num1=undefined, num2=undefined, operation=favicon.ico
2024-03-14 11:39:20 ERROR: Operation not found: notAnOperation
2024-03-14 11:39:20 ERROR: Invalid input: num1=undefined, num2=undefined, operation=favicon.ico
2024-03-14 11:39:44 ERROR: Error occurred: Cannot find the square root of a negative number.
2024-03-14 11:39:44 ERROR: Invalid input: num1=undefined, num2=undefined, operation=favicon.ico
2024-03-14 11:40:04 ERROR: Error occurred: Cannot divide by zero.
2024-03-14 11:40:04 ERROR: Invalid input: num1=undefined, num2=undefined, operation=favicon.ico
```