Water Jug Challenge

Overview:

In this exercise, backend candidates will design a solution for the classic Water Jug Riddle. The task is to create an API that can compute the steps required to measure exactly Z gallons using two jugs of capacities X and Y gallons. The backend should process this problem efficiently and return the solution steps in JSON format through a RESTful API.

Requirements:

1. **RESTful API:**
    - Develop endpoints to accept jug capacities (X and Y) and the target volume (Z) as inputs.
    - Ensure the API strictly adheres to REST principles with appropriate use of HTTP methods.
    - Responses must be provided in JSON format detailing the sequence of actions to solve the riddle or indicating if no solution is possible.
2. **Algorithm Implementation:**
    - Implement an algorithm that determines the sequence of steps needed to measure exactly Z gallons, if feasible.
    - The solution must consider the actions: Fill, Empty, and Transfer (between the two jugs only).
3. **Performance Considerations:**
    - Optimize the algorithm for quick response times, especially considering the potential for large input values.
4. **Error Handling and Validation:**
    - Validate input to ensure X, Y, and Z are positive integers.
    - Return meaningful error messages in JSON format if inputs are invalid or if no solution can be found.
5. **Testing:**
    - Create unit tests to verify the correctness of the algorithm.
    - Include integration tests to ensure the API handles requests and responses correctly.
6. **Documentation:**
    - Provide a README.md file detailing the API endpoints, how to set up and run the application, and an explanation of the algorithm used.
    - Include example requests and responses for clarity.
7. **Scalability extra points not necessary:**
    - Design the API to handle a high number of requests efficiently, using techniques such as caching results for common requests.

Deliverables:

- **Source Code:** Host the complete source code on a public GitHub repository.
- **API Documentation:** Detailed documentation of endpoints, including URI, request parameters, response format, and status codes and your algorithmic approach

- **README.md:** Instructions for setting up and running the backend, along with detailed documentation aimed at assessing written communication skills.

Sample Payload JSON: POST

{

“x_capacity”:2,

“y_capacity":10,

“z_amount_wanted”:4,

}

Sample Response JSON

{

"solution": [

{"step": 1, "bucketX": 2, "bucketY": 0, "action": "Fill bucket X"},

{"step": 2, "bucketX": 0, "bucketY": 2, "action": "Transfer from bucket X to Y"},

{"step": 3, "bucketX": 2, "bucketY": 2, "action": "Fill bucket X"},

{"step": 4, "bucketX": 0, "bucketY": 4, "action": "Transfer from bucket X to Y", "status": "Solved"}

]

}

Example:

Bucket X: 2

Bucket Y: 10

Amount wanted Z: 4

This is the best solution:

Bucket x	Bucket y	Explanation
2	0	Fill bucket x
0	2	Transfer from bucket x to bucket y
2	2	Fill bucket x
0	4	Transfer from bucket x to bucket y. **Solved**

Worst Solution

Bucket x	Bucket y	Explanation
0	10	Fill bucket y
2	8	Transfer from bucket y to bucket x
0	8	Empty bucket x
2	6	Transfer from bucket y to bucket x
0	6	Empty bucket x
2	4	Transfer from bucket y to bucket x. **SOLVED**

Example:

Bucket X: 2

Bucket Y: 100

Amount wanted Z: 96

This is the best solution:

Bucket x	Bucket y	Explanation
0	100	Fill bucket y
2	98	Transfer from bucket y to bucket x
0	98	Empty bucket x
2	96	Transfer from bucket y to x **SOLVED**

Worst Solution

Bucket x	Bucket y	Explanation
2	0	Fill bucket x
0	2	Transfer from bucket x to bucket y
2	2	Fill bucket x
0	4	Transfer from bucket x to bucket y
…	…	Repeat these steps a lot…..

0	96	Transfer bucket x to bucket y. **SOLVED**

Example:

Edge Case (No Solution)

Bucket X: 2

Bucket Y: 6

Amount Z: 5

Solution: No solution possible. Display “No Solution”.

Evaluation Criteria

Your submission will be evaluated based on the following criteria:

- Adherence to the technical requirements.
- Quality of the code and its organization.
- Documentation and ease of understanding your code and setup.