**Coding Exercise**

**Objective:**

To make sure our application is accessed by humans and not a by any computers or bot.

**Description:**

We want to make sure that the users who are using our application knows how to add numbers. To achieve this, we want to provide the user/client with a question and then allow the user/client to submit an answer. If the user/client and the service were both real person, then they would have interacted in the following manner:

Client: Hey Service, can you provide me a question with numbers to add?

Service: Here you go, solve the question: “Please sum the numbers 9,5,3”.

Client: Great. The original question was “Please sum the numbers 9,5,3” and the answer is 15.

Service: That’s wrong. Please try again.

Client: Sorry, the original question was “Please sum the numbers 9,5,3” and the answer is 17.

Service: That’s great

.

**Requirements:**

Create a server that does two things:

1. Respond to a client HTTP request

Respond to a client HTTP request with a minimum following in the response body

1. A question with random numbers in the response body.

(Some Sample questions could be:

Please sum the numbers 9,5,3

Please sum the numbers 10,2

Please sum the numbers 5,7,1)

1. Receive a client HTTP request

Receive a client HTTP request with a minimum following in the request

1. The question with the random numbers which was send earlier in the response of the first request.
2. The sum of the numbers in the question.

And respond with,

1. If the sum of the numbers is correct, then return a HTTP 200 OK
2. If the sum of the numbers is wrong or if it’s an invalid request, then return a HTTP 400 Bad Request.

You are not required to write the client application, but you can assume that a client for this server should:

1. Send a request to Service 1 and receives a response body with the question.
2. Sum the numbers in the question
3. Send the sum of the numbers to the service 2 along with the original response body from 1 which had the question.

**Note: Please attach a postman collection for testing the services with your deliverables.**

**Tests:**

Please include tests to automatically validate the correctness of your code. You can write unit tests that can validate the correctness of individual small pieces of code or functional test to verify the expected responses from the two endpoints. Whatever you choose to implement these tests, it must be automated and simple to run. It should not be manually driven steps to test and compare results.

**Important Design Considerations:**

1. Safeguard against cheating: When the server receives the question and the sum of the numbers, it should verify that the question and the answer it has received is previously given to a particular client request.
2. Stateless: Please remove as much “state” as possible from the server. Any new request to the server should not depend on the state of the previous request.

**Guideline:**

1. The problem may seem little ambiguous in some places but that is intentional to replicate a real software development requirement and it depends on your ability to make good assumptions or trade-offs.
2. Please check in your code to your public GitHub account and share with us.
3. It will be ideal if you could implement the problem using Java8 and Spring Boot. If not, at least use Java and any other framework which will allow you to write the services and run them.
4. Choose your own style, data structures, request/response format, etc. but please provide with following along with your code:

Since I have a Windows 7 laptop, please make sure that your service can run in Windows environment.

Please provide a windows batch script to start the server.

Also, let us know what is the default port the service is running and what & where to change if I want the services to run other than the default port. The services should serve a single path “/”.

1. A problem like this can be expanded and improved indefinitely which one can continue working forever improving little by little but that’s not the goal. This exercise should only take few hours but certainly not more than 8 hours. Please try to set a maximum amount of time to work on this project and submit whatever you accomplish by end of that time window. If you are not able to finish the exercise within that timeline based on your satisfaction, please explain everything that you would like to add in your “Readme” section.
2. Please treat this exercise as an actual project and write code as you would actually do in your work place.
3. Please make sure to include a detailed “Readme” section in your GitHub project explaining your work, assumptions and trade-offs.