

Documentation

Design:

Task:

Fizz buzz

Write a server application, which generates the [Fizz buzz](#) sequence with the following restrictions:

- The application must have a single endpoint.
- This endpoint accepts a single parameter: the last element of the sequence.
- The return value should be the Fizz buzz sequence in JSON. Please design a meaningful data structure.
- Implement tests. The type of testing (unit test, integration test, etc.) is your decision.
- Use the framework you are most familiar with.
- When you make design decisions, please document your reasoning behind your choice.
- Please document how to build, test, and run the application.
- Use git as VCS.

Please hand in:

- A link to the git repo
- The additional documentation (can be part of the repo)

Steps:

- understanding the task
- selecting programming language
- finding framework , both development and testing

Understanding the task:

For making a server application I chose socket programming. During my studies I learnt the basics of it. Also I have some experience with it in Python. As Java was the preferred language I did some research on Java socket programming and I was reassured that I can use my experiences with a bit of learning a new environment.

I decided to make three classes. One for the server, one for the client and one for the simulation. With this setup I can simulate receiving data, processing it, generating the sequence and writing it to a JSON file. Also I wrote out the sequence on the console as a side effect, the JSON file will be generated by the server as output.json.

Selecting programming language:

As it was asked I wrote the app in Java. I have worked in Java before, but I had to learn new ways to build my application. First I spent time on exploring Java socket programming, JSON handling. I did find a library for JSON, so I could work with it, also I found many documentation on sockets.

Finding framework:

I was looking for a framework, which will help me in fast development, project handling, provides testing abilities. For me it was IntelliJ IDEA. It was easy to learn, gave me the features I wanted, and also it supports Java programming.

Testing:

As for the testing, my main goal was to test the logic behind the server. If the sequence generating algorithm is tested well enough, then we made sure that we are not making false result to be written out. I made several test cases to see, how does it react for some edge cases and for some default cases.

User manual:

Building and Running the application:

-To be able to run the app, you must have at least Java SE Development Kit 18.0.1.1 installed on your computer.

-Then download FizzBuzz_Server.jar from the linked GitHub repository.

-Open CommandLine and get to the library which you downloaded the FizzBuzz_Server.jar.

-type: java -jar FizzBuzz_Server.jar and press enter.

Testing the application:

-To test the application, download the FizzBuzz_Server.zip from the linked GitHub repository.

-Then unpack it, and import it as a project into IntelliJ IDEA.

- Build and run the MyServerTest.java

-If the project cannot be imported, then start a new project.

-On your computer ,find your new projects src folder and copy everything into that from the FizzBuzz_Server's src folder.

-Do the same with the .idea/libraries folder

-Build and run MyServerTest.java