

Assignment Deliverables:

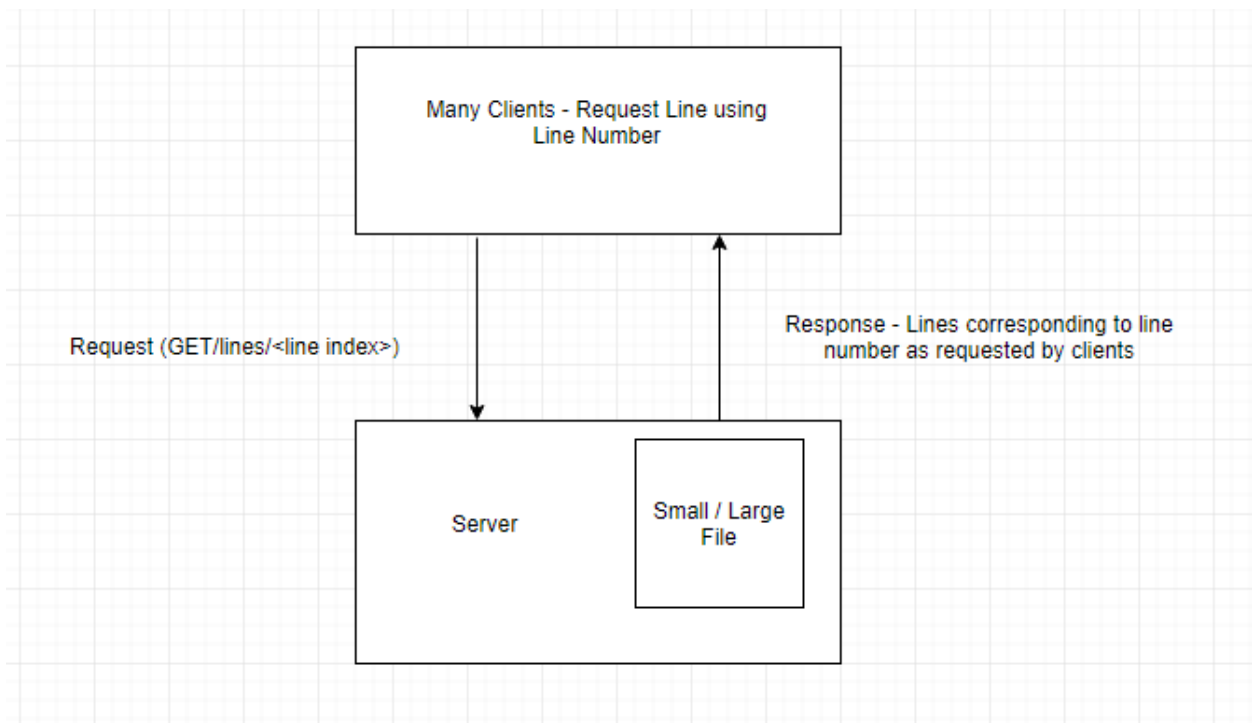
Contents

Repository (GitHub) Link for assignment	2
Use Case Diagram:	2
How does your system work? (If not addressed in comments in source)	2
What do we need to build your system?	2
How will your system perform with a 1 GB file? a 10 GB file? a 100 GB file?	8
How will your system perform with 100 users? 10000 users? 1000000 users?	8
What documentation, websites, papers, etc. did you consult in doing this assignment?	8
What third-party libraries or other tools does the system use? How did you choose each library or framework you used?	8
How long did you spend on this exercise? If you had unlimited more time to spend on this, how would you spend it and how would you prioritize each item?	9
If you were to critique your code, what would you have to say about it?	9
Test scripts in JUnit:	9

Repository (GitHub) Link for assignment

<https://github.com/mauryar/line-server-application>

Use Case Diagram:



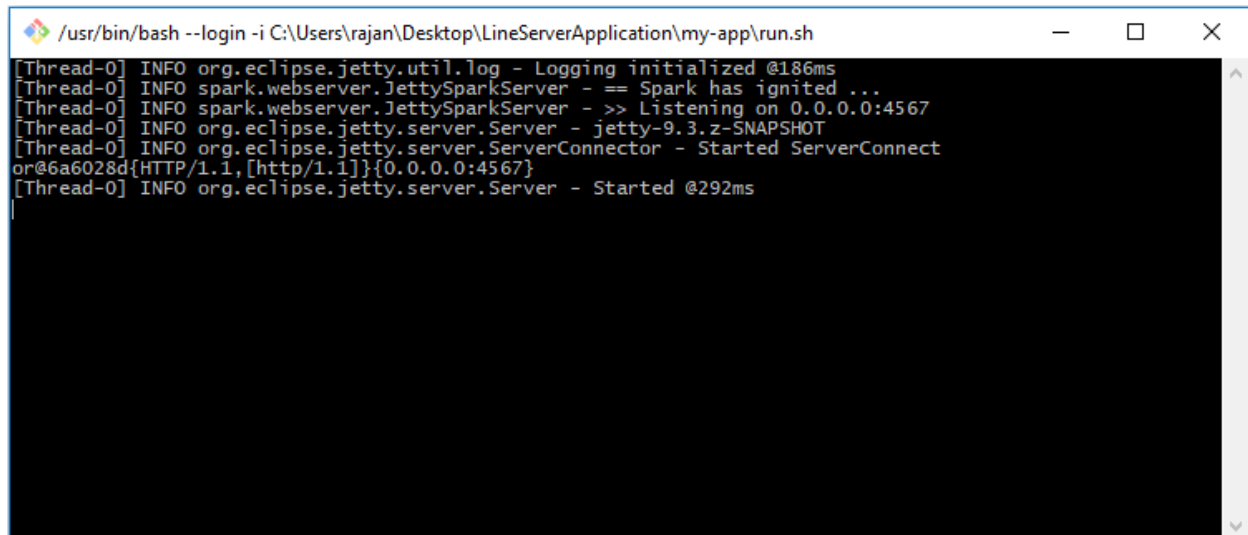
How does your system work? (If not addressed in comments in source)

Addressed in comments in source my-app-> src/main/java -> com.circle.controllers

What do we need to build your system?

- Download the source code from GitHub or unzip the Source code file
- Open my-app folder and then open the run.sh file in Notepad
- Enter the path of the input file that will give the line for corresponding line number

- Run build.sh file to create the build
- Run run.sh file



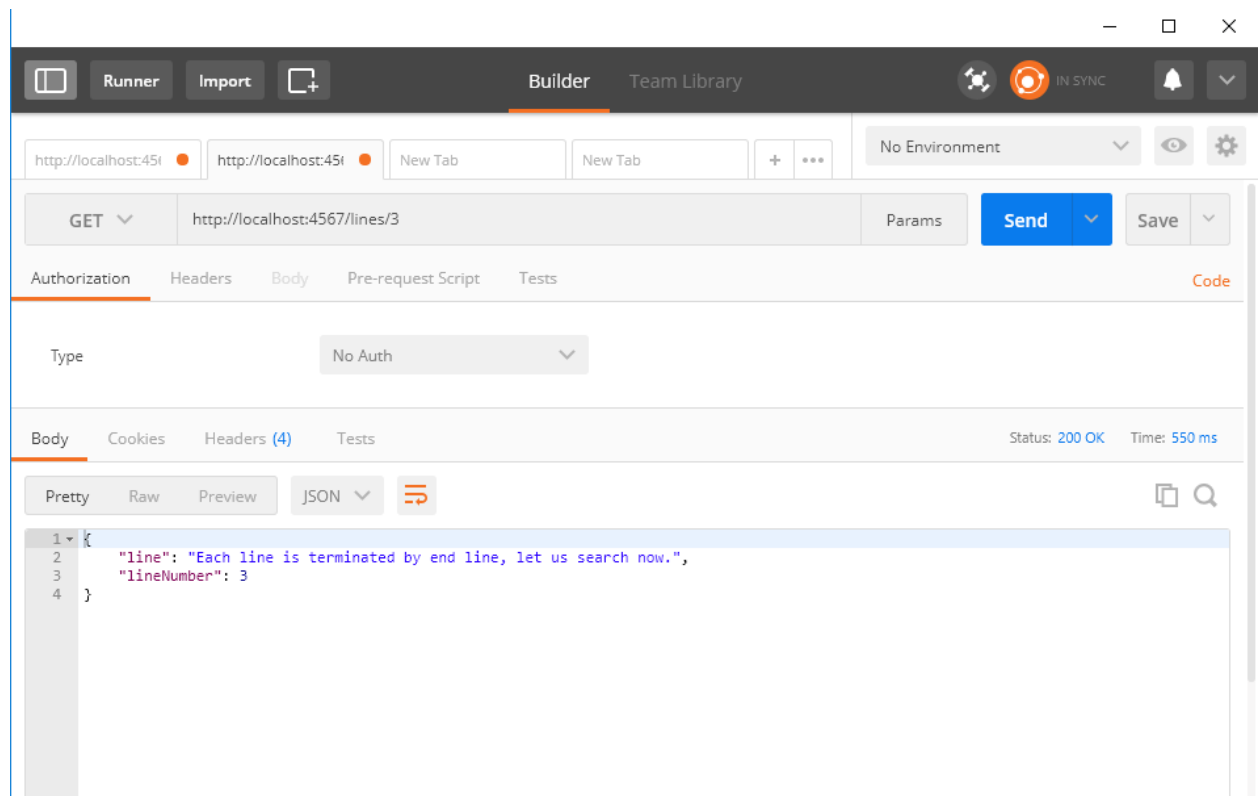
A screenshot of a Windows command prompt window. The title bar shows the path `/usr/bin/bash --login -i C:\Users\rajan\Desktop\LineServerApplication\my-app\run.sh`. The window contains the following output from a shell script:

```
[Thread-0] INFO org.eclipse.jetty.util.log - Logging initialized @186ms
[Thread-0] INFO spark.webserver.JettySparkServer - == Spark has ignited ...
[Thread-0] INFO spark.webserver.JettySparkServer - >> Listening on 0.0.0.0:4567
[Thread-0] INFO org.eclipse.jetty.server.Server - jetty-9.3.z-SNAPSHOT
[Thread-0] INFO org.eclipse.jetty.server.ServerConnector - Started ServerConnect
or@6a6028d{HTTP/1.1,[http/1.1]}{0.0.0.0:4567}
[Thread-0] INFO org.eclipse.jetty.server.Server - Started @292ms
```

- Use a REST Call on browser (I have used Postman)

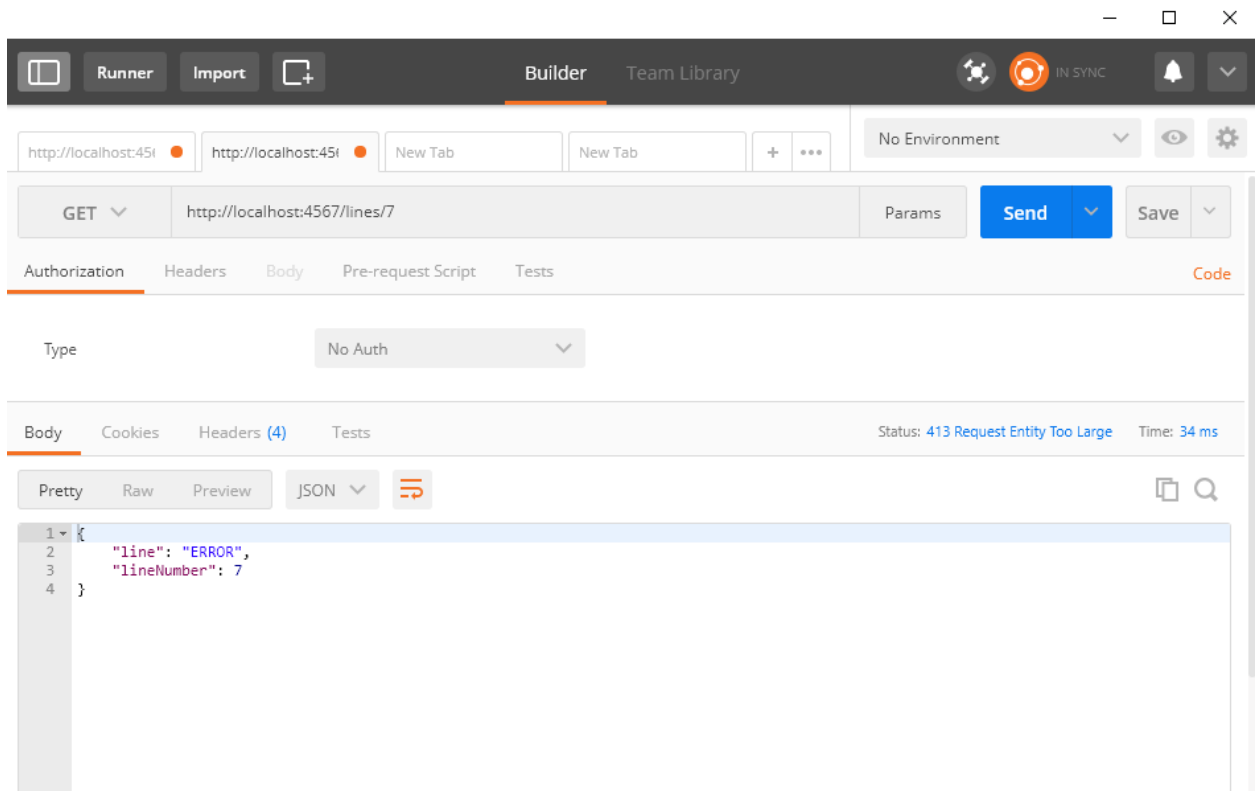
Example 1: <http://localhost:4567/lines/3>

Below is the screen print of the result with status code as 200



Example 2: <http://localhost:4567/lines/7>

Below is the screen print of the result with status code as 413 when line number exceeds the lines in file or '0' or negative line number



Example 3: http://localhost:4567/lines/0

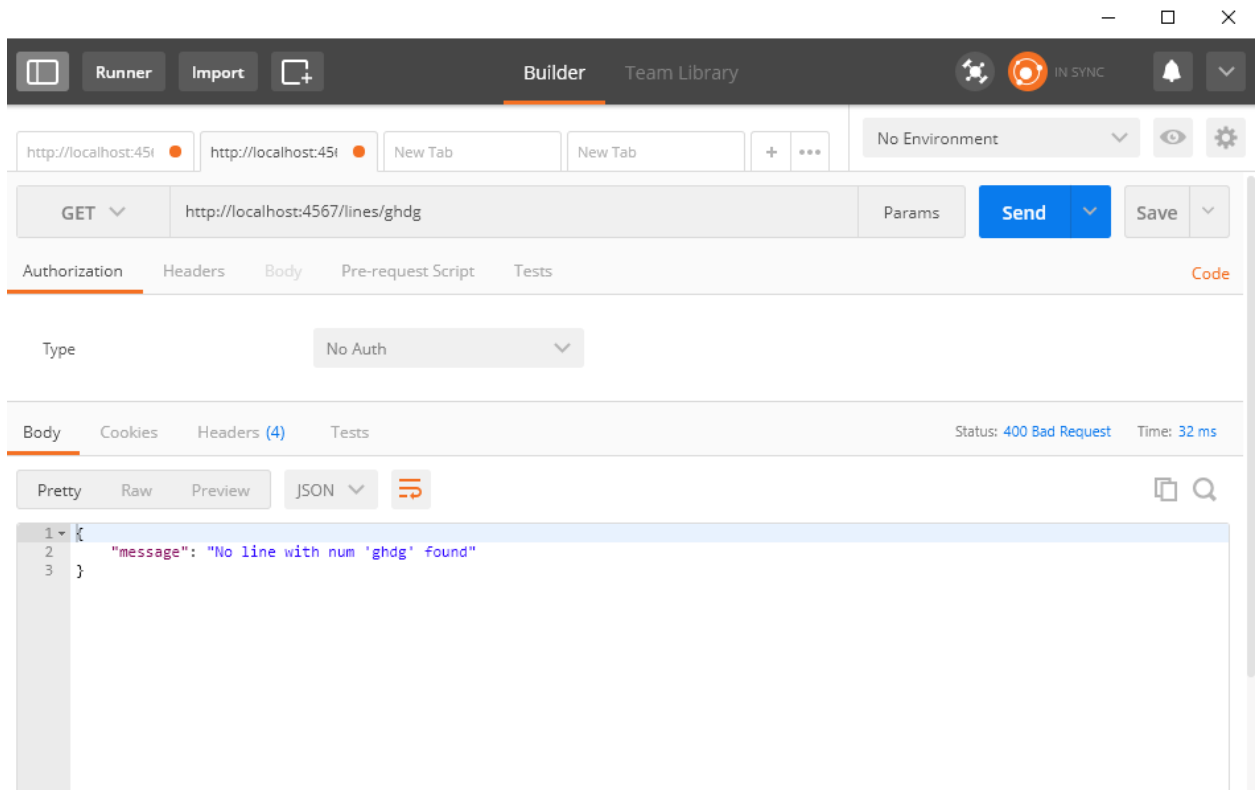
The screenshot displays the Postman API client interface. At the top, the 'Builder' tab is active. The URL bar shows 'http://localhost:4567/lines/0'. The method is set to 'GET'. The 'Send' button is highlighted in blue. Below the URL bar, the 'Authorization' tab is selected, showing 'No Auth'. The 'Body' tab is also visible. The response body is displayed in the 'Body' tab, showing a JSON object:

```
{
  "line": "ERROR",
  "lineNumber": 0
}
```

. The status bar at the bottom indicates 'Status: 413 Request Entity Too Large' and 'Time: 28 ms'.

Example 4: <http://localhost:4567/lines/ghdg>

Below is the screen print of the result with status code as 400 for bad input (alphabets as line number)



How will your system perform with a 1 GB file? a 10 GB file? a 100 GB file?

Tested successfully with 1GB file and system works well. However as the file size will increase, we can use Hadoop MapReduce to split the file and get the result.

How will your system perform with 100 users? 10000 users? 1000000 users?

Used 'synchronized' keyword for the '*getLineNumber()*' function for concurrent access from multiple users. Tested successfully for 1000 users using JMeter.

What documentation, websites, papers, etc. did you consult in doing this assignment?

<http://www.mscharhag.com/java/building-rest-api-with-spark>

<http://sparkjava.com/tutorials/>

<https://dzone.com/articles/building-simple-restful-api>

<http://www.cdn.geeksforgeeks.org/fast-io-in-java-in-competitive-programming/>

<https://github.com/arekgofi/Setting-up-Spark-with-Maven>

What third-party libraries or other tools does the system use? How did you choose each library or framework you used?

Chose Maven so that I need not download jars for the assignment.

Chose Java Spark as it is easy to implement and to get started with REST calls.

Chose Maven to build the project as it was easy to do with the maven project.

How long did you spend on this exercise? If you had unlimited more time to spend on this, how would you spend it and how would you prioritize each item?

I took around 8 hours to do this assignment.

If given more time (mentioned according to priority):

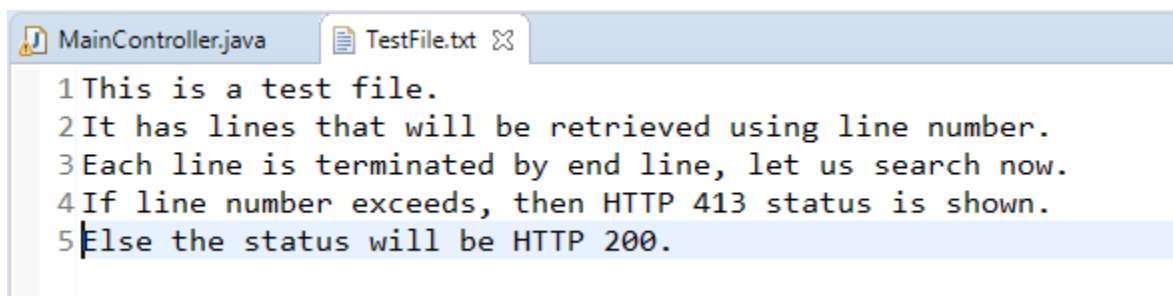
1. Handled more cases like when user does not give any line number
2. Use Jenkins
3. Make it work for huge files using Hadoop MapReduce
4. Create Stop script to kill the process
5. Use Swagger for REST Calls
6. Good UI to enter Line number

If you were to critique your code, what would you have to say about it?

It is good and could have been enhanced by implementing with Jenkins and a stop script.

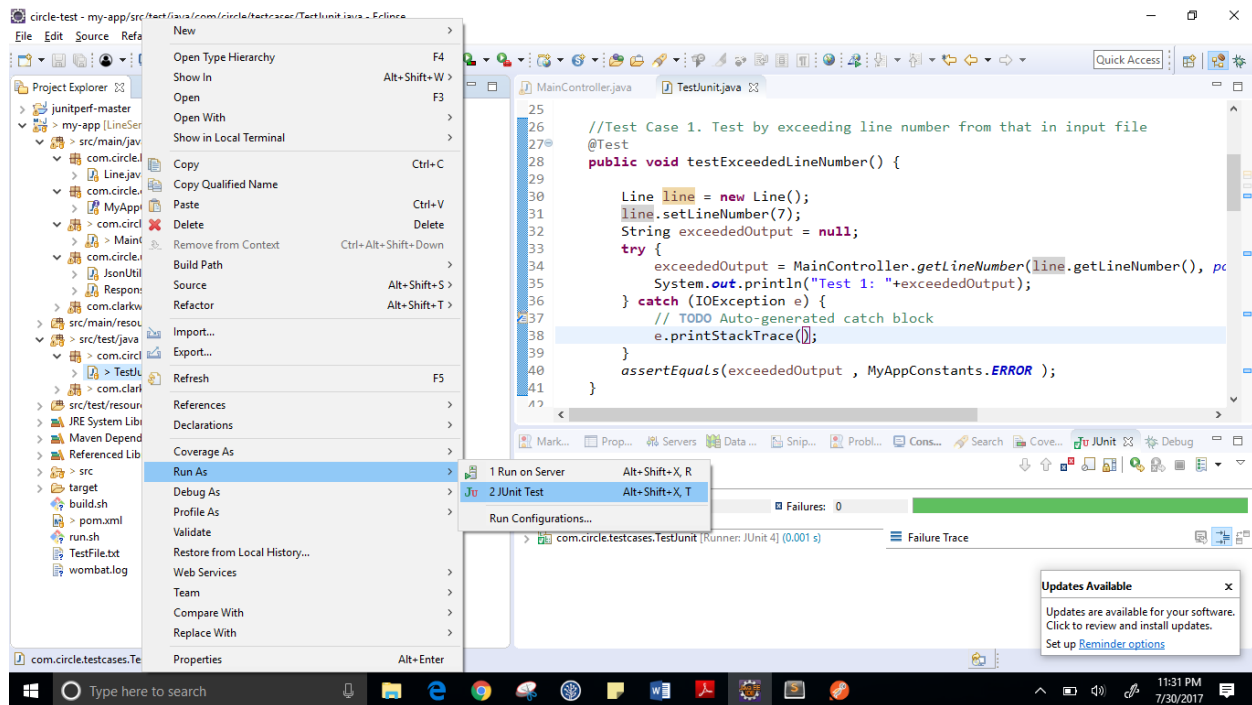
Test scripts in JUnit:

Input File: TestFile.txt



```
MainController.java TestFile.txt
1 This is a test file.
2 It has lines that will be retrieved using line number.
3 Each line is terminated by end line, let us search now.
4 If line number exceeds, then HTTP 413 status is shown.
5 Else the status will be HTTP 200.
```

Running Test cases:



Output:

