**----INSTRUCTIONS TO EXECUTE----**

**Software Needed**: Java 1.8x

**Software Recommended**: Spring Tool Suite (Eclipse or any other java IDE will work but STS is recommended).

In the zip file submitted, you can find the folder containing the code for Client (Java project), Server1, Server2, Server3 and Server4 (Maven Projects).

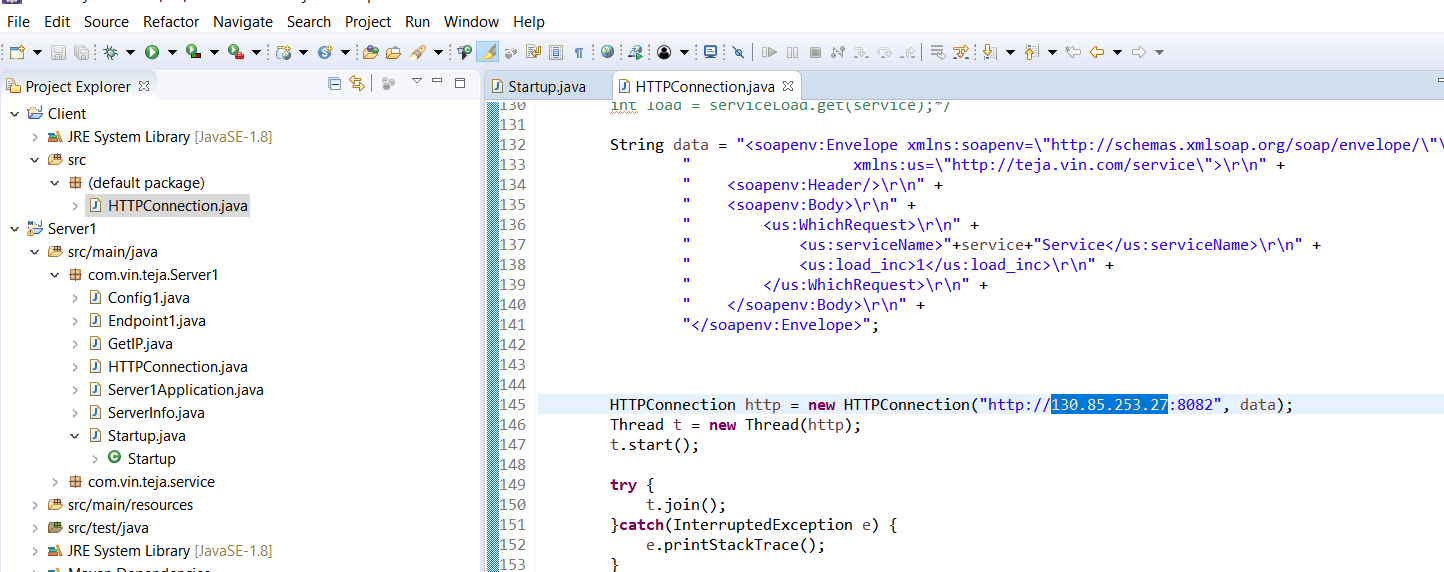
Along with these you can find a Quick Execute folder which contains the Jar files of the respective servers and also the client application java file. This should be used only if executing on multiple ports of the localhost on a single system.

**Scenario 1:** Using different localhost ports to run servers

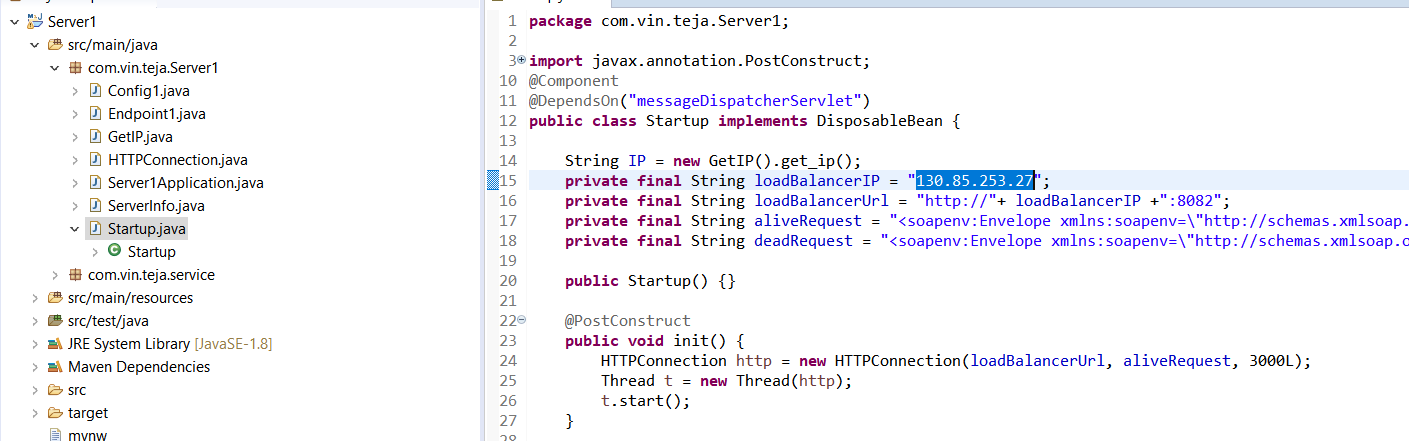
* If the plan is to execute on a single system across multiple ports of the localhost, then you can directly run the Jar files using the {java -jar jarFileName.jar} command to deploy the servers.
* It is mandatory to run Server1 before any of the other servers or clients. The order of the remaining servers/client does not matter.
* After the servers have been deployed, the client can be run by compiling and executing the HTTPConnection.java file using javac & java commands. This can be run multiple times to run service requests consecutively.

**Scenario 2:** Running servers on different machines

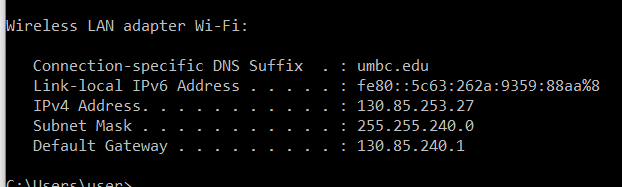
* In this case we must hardcode the IP and port of the machine that contains Server1 in 5 locations which are, the standalone HTTPConnection class of the client and in the startup classes of each server.
* To do this the server folders must be imported into the IDE as existing maven projects and select their directory to import them.
* The client folder can be opened normally using the open option in the IDE.



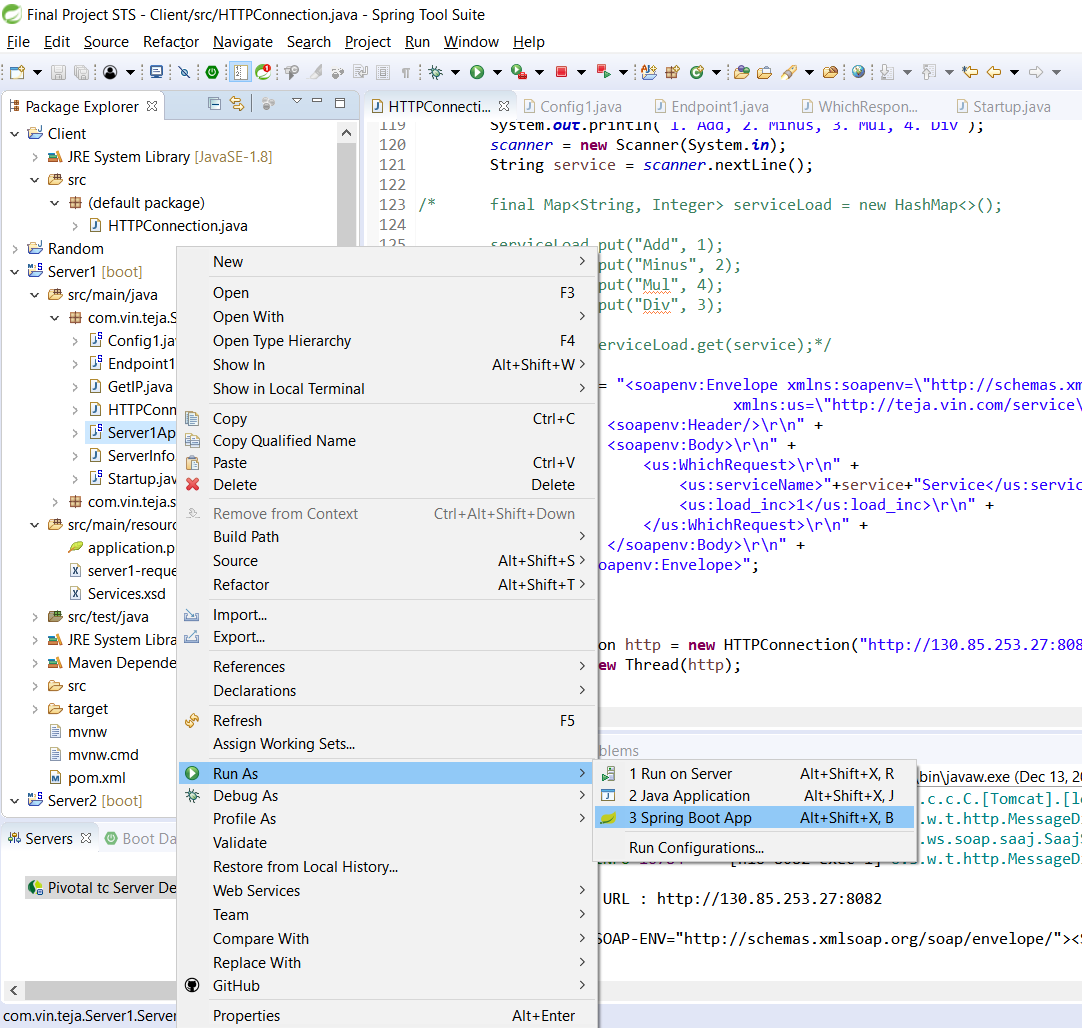
The above image shows the standalone **client HTTPConnection class**. Replace highlighted IP with **IPv4** of machine containing Server1.



This image is for the Startup classes **in each server**. Replace highlighted IP with **IPv4** of machine containing Server1.



Execution can be done by executing the servers as a spring boot app using Spring Tool Suite as follows:



Similar to Scenario 1, Server1 must be executed first. The rest of the servers can be executed in any order. Finally run HTTPConnection class of client to request the services by running it as a Java application.

If using the eclipse IDE follow the next steps: Run the server projects with the maven build option

.

.

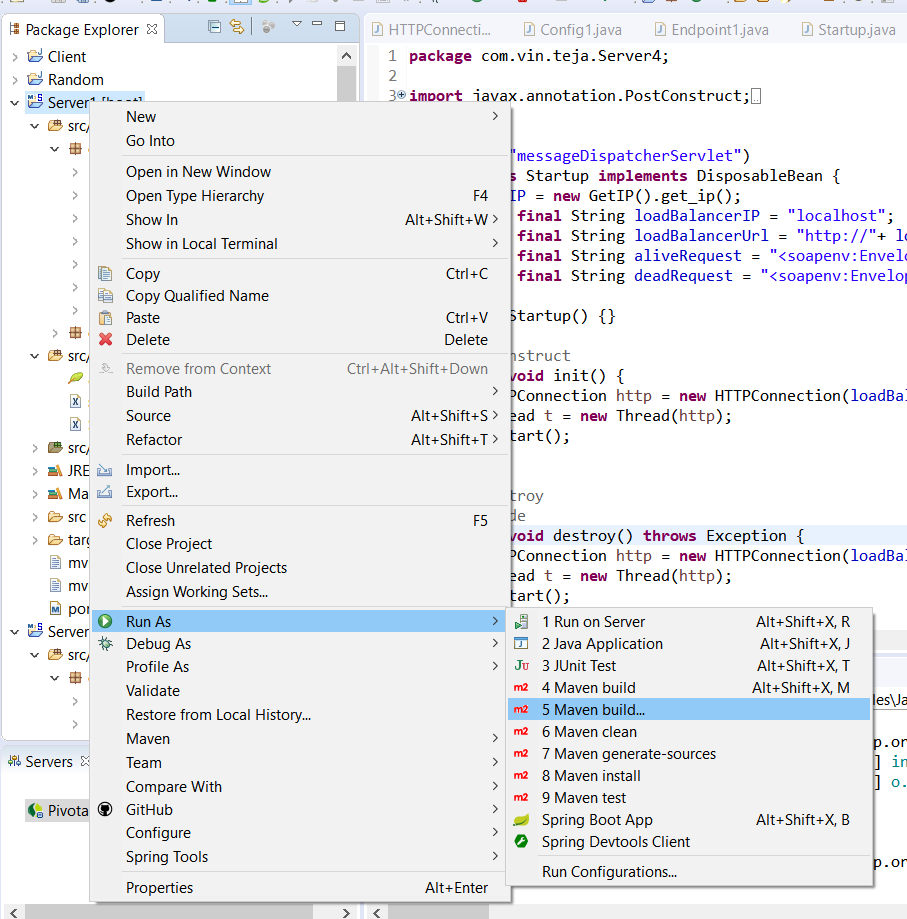
.

.

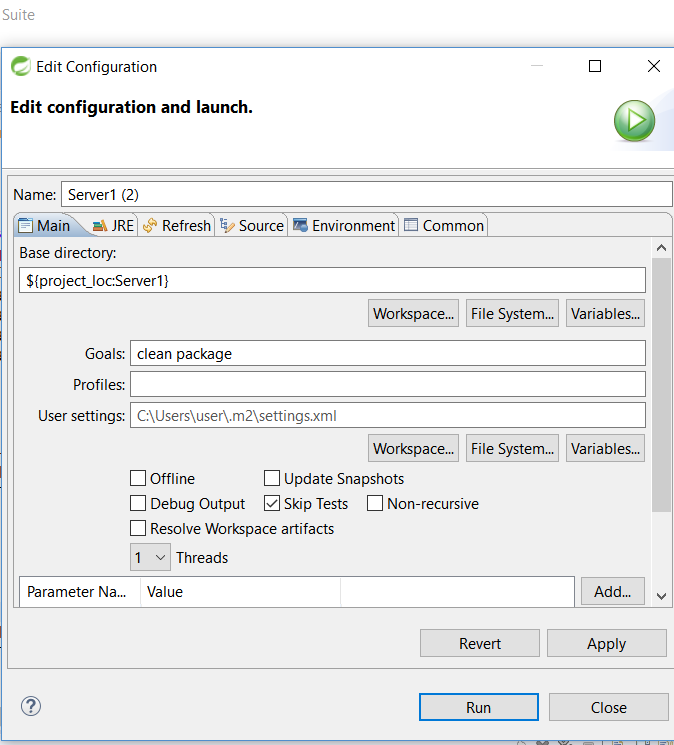
.

.

.



Next run it with the following configuration.



Make sure in the JRE option, the **Workspace Default JRE** is set to the local JDK path and run it to generate the jar file in the **target folder** of the particular project in the IDE. It can be run similarly using the {java –jar jarFileName.jar} command.