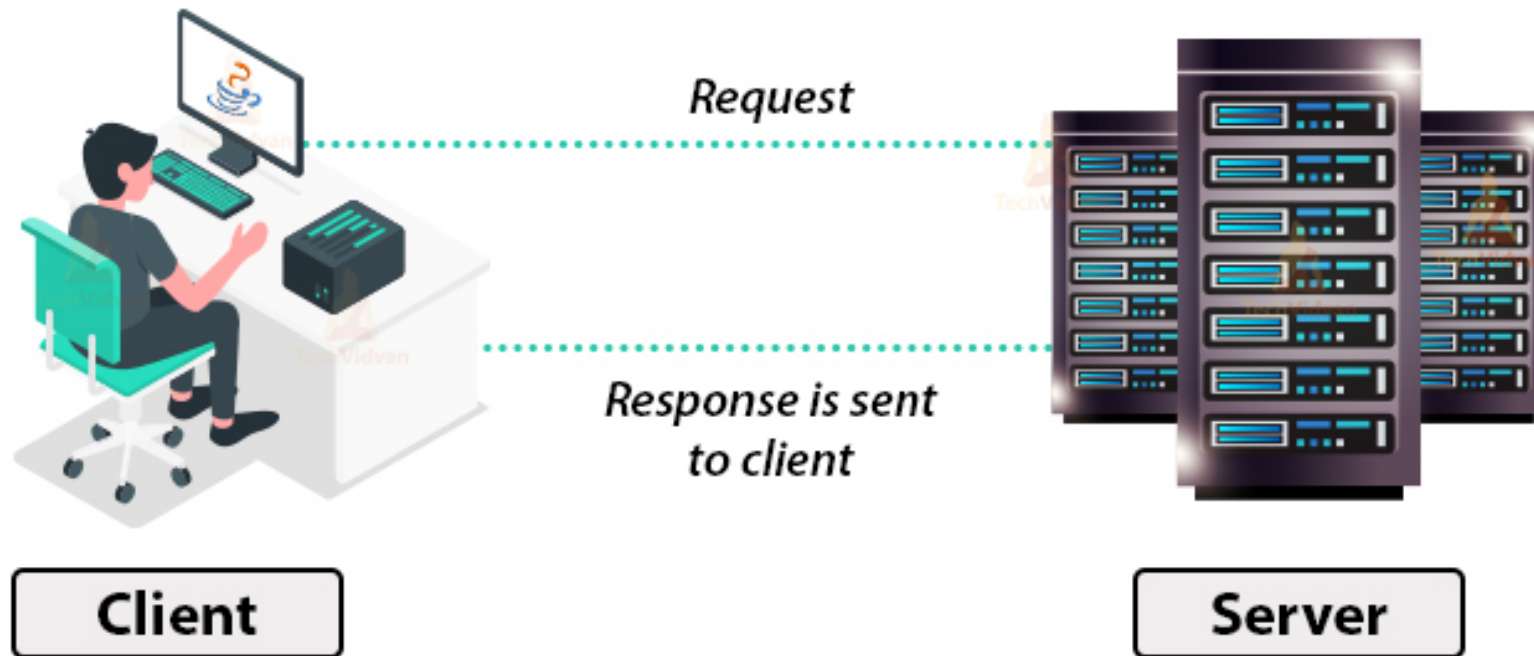


Socket Programming

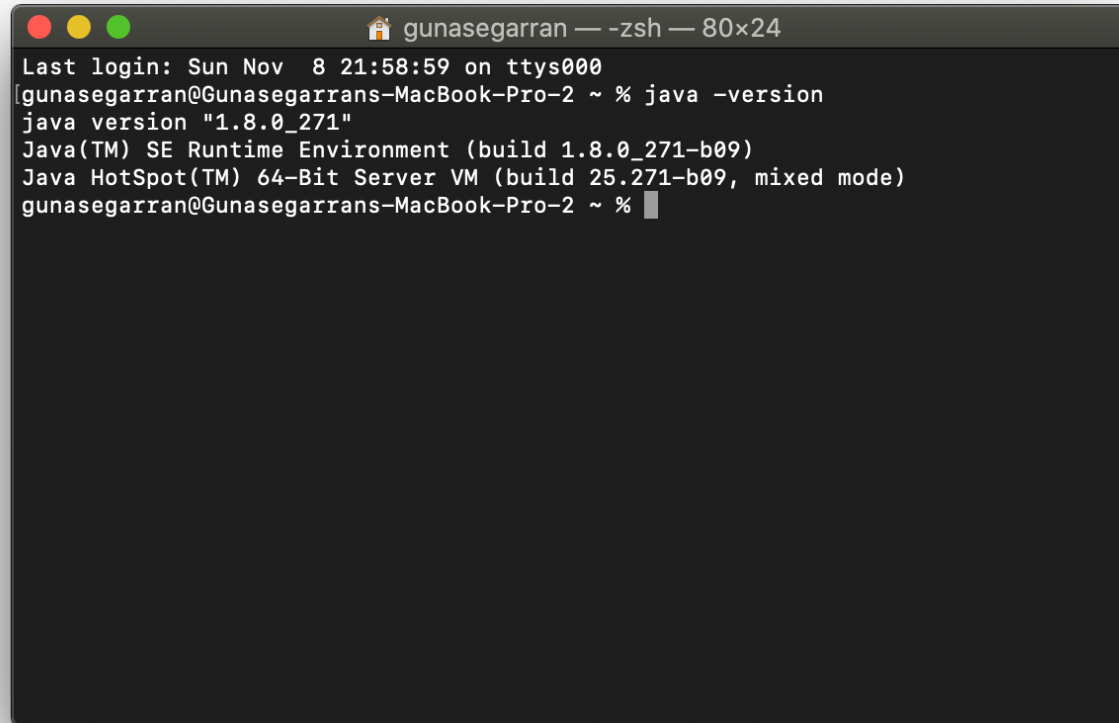




Java Socket Programming Process



1. Prerequisites JAVA is pre-installed

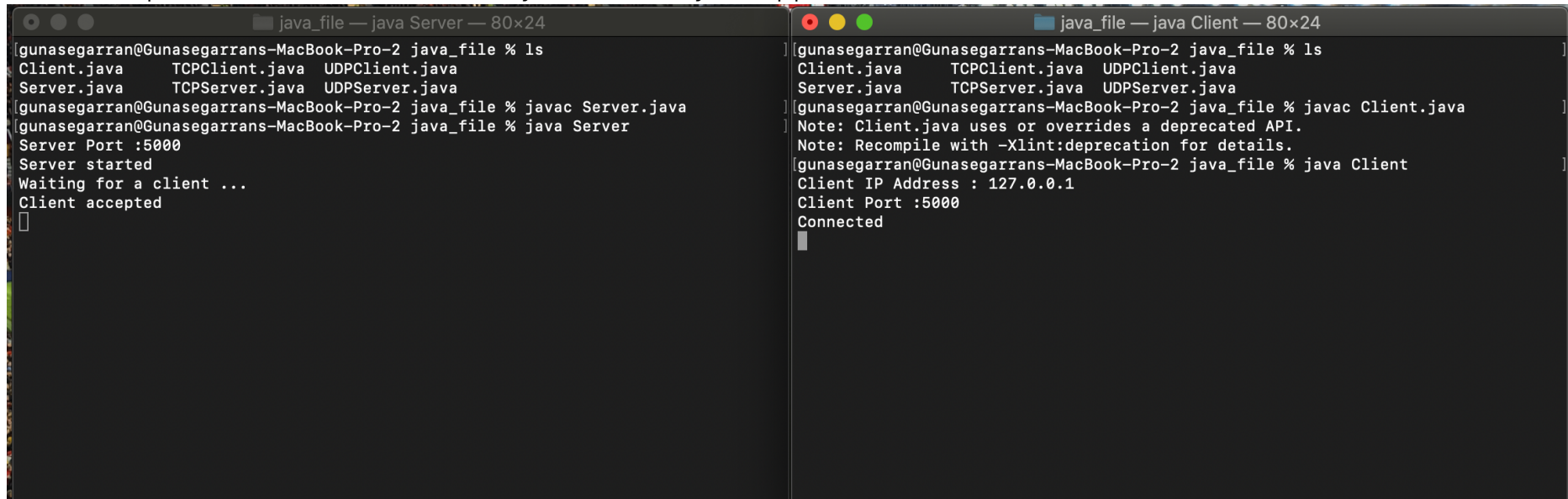


```
gunasegarran — -zsh — 80x24
Last login: Sun Nov  8 21:58:59 on ttys000
[gunasegarran@Gunasegarrans-MacBook-Pro-2 ~ % java -version
java version "1.8.0_271"
Java(TM) SE Runtime Environment (build 1.8.0_271-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.271-b09, mixed mode)
gunasegarran@Gunasegarrans-MacBook-Pro-2 ~ %
```



2. Testing Socket : Client - Server

A socket in Java is one endpoint of a two-way communication link between two programs running on the network. A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent to.



```
java_file — java Server — 80x24
gunasegarran@Gunasegarrans-MacBook-Pro-2 java_file % ls
Client.java      TCPClient.java  UDPClient.java
Server.java      TCPServer.java  UDPServer.java
gunasegarran@Gunasegarrans-MacBook-Pro-2 java_file % javac Server.java
gunasegarran@Gunasegarrans-MacBook-Pro-2 java_file % java Server
Server Port :5000
Server started
Waiting for a client ...
Client accepted
█

java_file — java Client — 80x24
gunasegarran@Gunasegarrans-MacBook-Pro-2 java_file % ls
Client.java      TCPClient.java  UDPClient.java
Server.java      TCPServer.java  UDPServer.java
gunasegarran@Gunasegarrans-MacBook-Pro-2 java_file % javac Client.java
Note: Client.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
gunasegarran@Gunasegarrans-MacBook-Pro-2 java_file % java Client
Client IP Address : 127.0.0.1
Client Port :5000
Connected
█
```

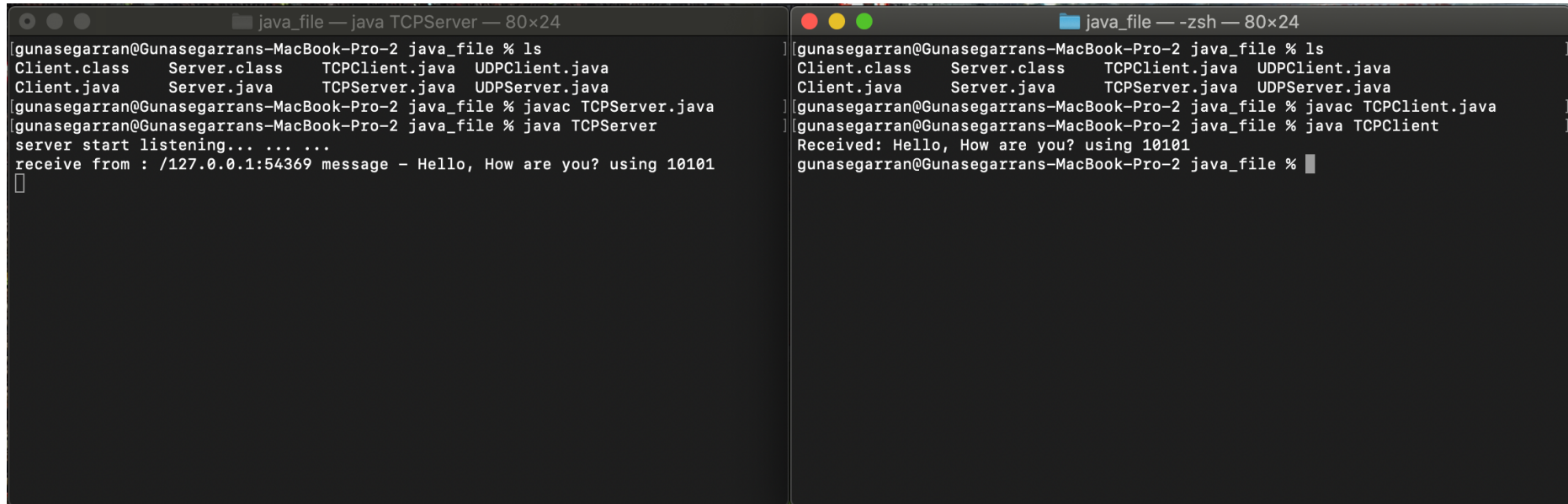
The very first step is to establish a socket connection. A socket connection implies that the two machines have information about each other's network location (IP Address) and TCP port.

Socket socket = new Socket("127.0.0.1", 5000)

- Here, the first argument represents the **IP address of Server**.
- The second argument represents the **TCP Port**. (It is a number that represents which application should run on a server.)



3. Testing TCP Connection Client Server



```
java_file — java TCPServer — 80x24
[gunasegarran@Gunasegarrans-MacBook-Pro-2 java_file % ls
Client.class  Server.class  TCPClient.java  UDPClient.java
Client.java   Server.java   TCPServer.java  UDPServer.java
[gunasegarran@Gunasegarrans-MacBook-Pro-2 java_file % javac TCPServer.java
[gunasegarran@Gunasegarrans-MacBook-Pro-2 java_file % java TCPServer
server start listening... ..
receive from : /127.0.0.1:54369 message - Hello, How are you? using 10101
]

java_file — -zsh — 80x24
[gunasegarran@Gunasegarrans-MacBook-Pro-2 java_file % ls
Client.class  Server.class  TCPClient.java  UDPClient.java
Client.java   Server.java   TCPServer.java  UDPServer.java
[gunasegarran@Gunasegarrans-MacBook-Pro-2 java_file % javac TCPClient.java
[gunasegarran@Gunasegarrans-MacBook-Pro-2 java_file % java TCPClient
Received: Hello, How are you? using 10101
gunasegarran@Gunasegarrans-MacBook-Pro-2 java_file % ]
```

A TCP connection is established with the help of three-way handshake. It is a process of initiating and acknowledging a connection. Once the connection is established, data transfer begins, and when the transmission process is finished, the connection is terminated by the closing of an established virtual circuit.



4. Testing UDP Connection Client Server

```
java_file — java UDPSever — 80x24
[gunasegarran@Gunasegarrans-MacBook-Pro-2 java_file % ls
Client.class      Server.java      TCPSever.java
Client.java       TCPClient.class  UDPClient.java
Connection.class  TCPClient.java   UDPSever.java
Server.class      TCPSever.class
[gunasegarran@Gunasegarrans-MacBook-Pro-2 java_file % javac UDPSever.java
[gunasegarran@Gunasegarrans-MacBook-Pro-2 java_file % java UDPSever
start server...
receive from : /127.0.0.1:53991 message - Hello World
reply to    : /127.0.0.1:53991
[]

java_file — -zsh — 80x24
[gunasegarran@Gunasegarrans-MacBook-Pro-2 java_file % ls
Client.class      Server.java      TCPSever.java
Client.java       TCPClient.class  UDPClient.java
Connection.class  TCPClient.java   UDPSever.java
Server.class      TCPSever.class
[gunasegarran@Gunasegarrans-MacBook-Pro-2 java_file % javac UDPClient.java
[gunasegarran@Gunasegarrans-MacBook-Pro-2 java_file % java UDPClient
start client...
send to      : localhost/127.0.0.1:10101 message - Hello World
waiting server reply ...
reply from   : /127.0.0.1:10101 message - Hello World
gunasegarran@Gunasegarrans-MacBook-Pro-2 java_file %
```

UDP uses a simple transmission method without implied hand-shaking dialogues for ordering, reliability, or data integrity. UDP also assumes that error checking and correction is not important or performed in the application, to avoid the overhead of such processing at the network interface level. It is also compatible with packet broadcasts and multicasting.

