**Overview**

This Java application consists of a JokeServer and multiple JokeClient programs that communicate over TCP/IP to exchange jokes. The server can handle multiple clients simultaneously, each interacting in separate processes.

**Requirements**

* Java Version: 19.0.1
* Files: JokeServer.java, JokeClient.java

**Compilation**

To compile the programs, run the following command twice in the terminal:

sh

Copy code

> javac \*.java

**Running the Application**

**On Localhost**

1. Open multiple terminal windows.
2. In the first terminal, start the server:

sh

Copy code

> java JokeServer

1. In the remaining terminals, start the clients:

sh

Copy code

> java JokeClient

**Over the Internet**

1. Open multiple terminal windows.
2. In the first terminal, start the server:

sh

Copy code

> java JokeServer

1. In the remaining terminals, start the clients with the server's IP address:

sh

Copy code

> java JokeClient <Server-IP>

**How It Works**

1. **JokeClient**: The client prompts the user to enter their name and request a joke or proverb.
   * The client establishes a TCP/IP connection to the server.
   * The request is sent to the server and a response is received.
   * The connection is terminated after each joke is received.
   * The client keeps track of the number of jokes received.
2. **JokeServer**: The server listens for incoming client connections.
   * Each client request is handled by a separate JokeWorker.
   * The server responds with a random joke.
   * The server keeps track of the total number of jokes served to each client.

**Features**

* Multiple clients can connect to the server simultaneously.
* The server maintains a separate conversation with each client.
* The application uses TCP/IP for reliable communication.
* Jokes are sent as serialized Java objects over the network.
* The server tracks the number of jokes given to each client.

**Usage**

* **Requesting a Joke**: Enter your name when prompted, and then request a joke by typing in the terminal.
* **Stopping the Client**: Type "quit" to stop the client.

**Notes**

* The server and client communicate using a connectionless, stateless protocol.
* The server can track and count jokes, making it potentially useful for other applications like transaction tracking or sales records.
* Typing in the code yourself can help you understand multi-process, multi-threaded network programming better.

**Acknowledgments**

* [Sending objects over sockets - Java example](https://www.comrevo.com/2019/07/Sending-objects-over-sockets-Java-example-How-to-send-serialized-object-over-network-in-Java.html) by Ramesh
* [Java SocketException](https://rollbar.com/blog/java-socketexception/)
* Hughes, Shoffner, and Winslow for Inet code.

**Example Output**

vbnet

Copy code

To Request a Joke you need to give your name:

To Request a Proverb you need to give your name:

Hi <UserName>

Choose a Joke you want:

<joke>

RESPONSE RECEIVED:

<response>

The Joke sent back is: <joke>

The Joke count is: <count>

Closing the connection to the server.

Cancelled by user request.

<UserName>, You sent and received <count> Jokes.

This README provides a comprehensive overview of the Joke Server and Client application, detailing how to compile, run, and use the programs, along with the features and functionalities they offer.