```java
import io.opentelemetry.api.GlobalOpenTelemetry;
import io.opentelemetry.api.trace.Span;
import io.opentelemetry.api.trace.Tracer;
import io.opentelemetry.api.trace.TracerProvider;
import io.opentelemetry.context.Scope;

import java.io.*;
import java.net.Socket;
import java.util.UUID;

public class client {
    private static final Tracer tracer = GlobalOpenTelemetry.getTracer("client-tracer");

    public static void main(String[] args) {
        System.setProperty("otel.tracer.provider", "io.opentelemetry.api.trace.propagation.B3Propagator$Factory");

        try {
            System.out.println("Connecting to the server...");

            // Create a socket and connect to the server on localhost, port 8080
            Socket socket = new Socket("localhost", 8080);

            System.out.println("Connected to the server.");

            // Create input and output streams for communication with the server
            BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
            PrintWriter out = new PrintWriter(socket.getOutputStream(), true);

            // Specify the folder path on the client side
            String desktopPath = System.getProperty("
```

```
30 user.home") + "/Desktop" + "/Files";
31             File folder = new File(desktopPath);
32 
33             // List files in the folder
34             File[] files = folder.listFiles();
35             if (files == null) {
36                 System.err.println("No files found in
    the folder: " + desktopPath);
37                 return;
38             }
39 
40             // Send up to 20 files at a time
41             int filesToSend = Math.min(20, files.
   length);
42             out.println(filesToSend);  // Send the
   number of files to expect
43             for (int i = 0; i < filesToSend; i++) {
44                 File file = files[i];
45 
46                 // Send the file name to the server
47                 out.println(file.getName());
48 
49                 // Read the file content and send it
   to the server
50                 try (BufferedReader fileReader = new
   BufferedReader(new FileReader(file))) {
51                     String line;
52                     while ((line = fileReader.
   readLine()) != null) {
53                         out.println(line);
54                     }
55                 }
56 
57                 System.out.println("File sent: " +
   file.getName());
58             }
59 
60             // Close the streams and socket
61             in.close();
62             out.close();
63             socket.close();
```

```
64
65          } catch (IOException e) {
66              System.err.println("Connection failed.
    Make sure the server is running and check your
    firewall settings.");
67              e.printStackTrace();
68          }
69      }
70 }
71
```