```
package modules.email.action;

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

public class GetEmail {

    public final static String OUTPATH = "store/email/outbox/";
    public final static String INPATH = "incoming/email";

    public static void getEmail (String name, String domain, String password){
        //Get Email Messages from Server, Save to incoming folder

        Socket server = null;
        BufferedWriter out = null;
        BufferedReader in = null;

        try {
            System.out.println("Opening socket...");
            server = new Socket("localhost", 20002);
            System.out.println("Connection successful!");
            out = new BufferedWriter(new OutputStreamWriter(server.getOutputStream()));
            in = new BufferedReader(new InputStreamReader(server.getInputStream()));

            System.out.println("Sending login information to the server.");

            if(server.isConnected()) {
                out.write("<?xml version='1.0'?>");
                out.write("<verify>");
                out.write("<domain>"+domain+"</domain>");
                out.write("<name>"+name+"</name>");
                out.write("<password>"+password+"</password>");
                out.write("</verify>");
                out.newLine();
                out.flush();    // flush should write
                server.shutdownOutput();

                int count = 1;
                String response = "";
                while(!(response = in.readLine()).contains("END")) {
                    if(response != null){
                        System.out.println(response);
                    try{
                        FileWriter fstream = new FileWriter("incoming/email/msg_"+count+".xml");
                        BufferedWriter fout = new BufferedWriter(fstream);
                        fout.write(response);
                        fout.close();
                        count++;
                    } catch (Exception e){
                        System.err.println("Error: "+e.getMessage());
                    }
                    }

                }   // end while loop
            } else {
                System.out.println("Connection with server could not be established.");
            }   // end if-else

            //out.close();
            in.close();
            server.close();

        } catch(Exception e) {
            e.printStackTrace();
        }

       try{
```

```
            Thread.sleep(2000);
        } catch (InterruptedException e){
             e.printStackTrace();
        }

         //Parse the imported files Originally found in the shell script



        int datecnt = 1;
        File folder = new File(INPATH);
        folder.mkdirs();
        File[] listOfFiles = folder.listFiles();
        for (File f : listOfFiles){
             if(f.isFile() && !f.isHidden()){
                 //Build the ACL2 script
                 String unique = (new Date().toString()) +"_" + datecnt;
                 unique = unique.replace(' ', '_');
                 unique = unique.replace(':', '_');
                 String script = "(in-package \"ACL2\")(include-book \"modules/email/action/rw-email\"" +
                         " :uncertified-okp t) (readEmail \"incoming/email/"+f.getName()+"\" \""+unique+"\"  ↙
    state)";

                 try{
                 //Run on ACL2
                 // Initialize ACL2 and dump its output to the log
                 System.out.println("Executing ACL2 runtime for Email Generation...");
                 ProcessBuilder processBuilder = new ProcessBuilder("acl2");
                 File log = new File("logs/acl2_log.txt");
                 processBuilder.redirectErrorStream(true);
                 processBuilder.redirectOutput(log);

                 Process process;

                 process = processBuilder.start();

                 PrintWriter procIn = new PrintWriter(process.getOutputStream());

                 // Write the ACL2 to the process, close ACL2
                 procIn.println(script);
                 procIn.println("(good-bye)");
                 procIn.flush();
                 procIn.close();

                 } catch(IOException e) {
                     e.printStackTrace();
                 }
             }

                try{
                    Thread.sleep(3000);
                } catch (InterruptedException e){
                     e.printStackTrace();
                }

             f.delete();
             datecnt ++;
        }




    }
}
```