

```

package modules.email.action;

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

/**
 * This file will replace the send-message.sh shell script and be included in the ACL2 email client
 * package.
 * @author Wesley R. Howell
 *
 * Original header:
 * #Created By: Wesley R. Howell
 * #Shell Script to Process an email message
 * #invocation ./route-email.sh "inputFile"
 * where inputFile is the email message in the
 * incoming/email folder on the server
 *
 */

public class SendEmail {

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

    /**
     * Function that replaces Shell Script for Sending Emails.
     * @param to
     * @param from
     * @param sub
     * @param msg
     */
    public static void sendMessage(String to, String from, String sub, String msg){
        //Build the ACL2 script
        String script = "(in-package \"ACL2\")(include-book \"modules/email/action/rw-email\" +
            \" :uncertified-okp t) (writeMessage \"\"+to+\"\" \"\"+from+\"\" \"\"+sub+\"\" \"\"+msg+\"\" 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(4000);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}

```

```

}
//Split Emails with new Java file
File folder = new File(OUTPATH);
File[] listOfFiles = folder.listFiles();

for (File f : listOfFiles){
    if(f.isFile() && !f.isHidden()){
        //Open the file and get contents.
        ArrayList<String> contents = new ArrayList<String>();
        StringBuilder strb = null;
        BufferedReader reader = null;

        try {
            //Use the string builder to append each email to the string,
            //Add the string to the list when it reaches the end
            //Start a new string builder for each new email in the file.
            reader = new BufferedReader(new FileReader (f));
            String line = null;

            try {
                while ((line = reader.readLine()) != null){
                    if(line.contains("<?xml version=")){
                        if(strb != null){
                            contents.add(strb.toString());
                        }
                        strb = new StringBuilder();
                    }
                    strb.append(line);
                }
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }

        } catch (FileNotFoundException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        contents.add(strb.toString());

        //Remove the file since we do not need it anymore.
        f.delete();

        int count = 1;

        for(String str : contents){

            /*
             We will now send each email to the server
            */

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

            try {
                System.out.println("Opening socket...");
                server = new Socket("localhost", 20005);
                System.out.println("Connection successful!");
                out = new PrintWriter(server.getOutputStream(), true);
                in = new BufferedReader(new InputStreamReader(server.getInputStream()));

                out.println(str);

                out.flush();
                out.close();
            }

```

```
        in.close();
        server.close();

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

}

}
```