

Verteilte Systeme und Kommunikationsnetze

Testumgebung

Praktikum 3

Fachhochschule Bielefeld
Campus Minden
Studiengang Informatik

Beteiligte Personen:

Name	Matrikelnummer
Michael Nickel	1120888
Jan Augstein	1119581

Aufgaben:

Aufgabe	Gelöst
Aufgabe 1	alle Teilpunkte
Aufgabe 2	alle Teilpunkte

19. November 2017

Inhaltsverzeichnis

1	Aufgabe 2	3
1.1	Frage- bzw. Aufgabenstellung	3
1.2	Vorbereitung	3
1.3	Aufgabenteil a)	3
	Frage- bzw. Aufgabenstellung	3
	Lösung	3
	Ergebnis	4
1.4	Aufgabeteil b)	4
	Frage- bzw. Aufgabenstellung	4
	Lösung	4
	Ergebnis	4
1.5	Aufgabenteil c)	5
	Frage- bzw. Aufgabenstellung	5
	Lösung	5
2	Quellen	5

1 Aufgabe 2

1.1 Frage- bzw. Aufgabenstellung

Schreiben Sie ein eigenes Java-Programm, das Mails versendet. Verwenden Sie dazu die Bibliothek javamail (javamail.java.net).

1.2 Vorbereitung

Zur Vorbereitung muss die jeweilige Bibliothek installiert werden und es muss sich in die Bibliothek eingearbeitet/ eingelesen werden.

1.3 Aufgabenteil a)

Frage- bzw. Aufgabenstellung

Verwenden Sie als SMTP Ausgangsserver einen Ihrer eigenen Mailaccounts, z.B. den der FH Bielefeld. Die Konfiguration kennen Sie aus den Hilfeseiten der DVZ. Das Programm soll in der Lage sein, eine Mail mit einem von ihnen vorgegebenen Inhalt an eine beliebige Mailadresse zu senden. Der Inhalt der Mail soll aus einer Datei inhalt.txt aus dem Dateisystem Ihres Rechners eingelesen werden.

Lösung

Damit das Programm E-Mails verschicken kann, müssen vorher Konfigurationen (Properties) festgelegt werden. Diese bestimmen das verwendete Protokoll (siehe Zeile 2), den Host-Server (siehe Zeile 3), den verwendeten Port (siehe Zeile 4), ob eine Authentifizierung stattfinden soll (siehe Zeile 5) und dass TLS benutzt wird (siehe Zeile 6).

```
1 | Properties props = new Properties();
2 | props.setProperty("mail.transport.protocol", "smtp");
3 | props.setProperty("mail.smtp.host", "smtp.fh-bielefeld.de");
4 | props.setProperty("mail.smtp.port", "587");
5 | props.setProperty("mail.smtp.auth", "true");
6 | props.setProperty("mail.smtp.starttls.enable", "true");
```

Zur Authentifizierung wird folgendes benötigt. Zum einen muss man die Session mit den zuvor bestimmten Konfigurationen erzeugen. Des Weiteren fügt man der Session einen Authenticator hinzu, der zur Abfrage des Users und des Passwortes ist (siehe Zeile 2-3).

```
1 | Session session = Session.getInstance(props, new
2 |     javax.mail.Authenticator() {
3 |         protected PasswordAuthentication
4 |             getPasswordAuthentication() {
5 |                 return new PasswordAuthentication("user",
6 |                     "password");
7 |             }
8 |     });
```

Zum Erstellen der E-Mail wird ein Objekt von Typ Message erstellt (Zeile 1). Diesem Objekt gibt man dann die EMail des Sender (Zeile 2) und des Empfängers (Zeile 3), einen Betreff (Zeile 4) und einen Inhalt (Zeile 5). Zum Senden der E-Mail benutzt man den Befehl Transport.send (siehe Zeile 7).

```

1 | Message message = new MimeMessage(session);
2 | message.setFrom(new InternetAddress(Sender Email));
3 | message.setRecipients(Message.RecipientType.TO,
   |     InternetAddress.parse(Empfaenger Email));
4 | message.setSubject("Betreff");
5 | message.setText("Inhalt");
6 |
7 | Transport.send(message);

```

Da der Inhalt aus eine Text-Datei genommen werden soll, muss man diese vorher einlesen(Zeile 1-2). Der String der durch das Lesen der Datei entsteht wird dann der jeweiligen Methode zum setzen des E-Mail Inhaltes übergeben(Zeile 4).

```

1 | ReadFile contentReader = new ReadFile();
2 | String mailText = contentReader.readFile("C:/Users/Jan
   |     Augstein/eclipse-workspace/VSPraktikum3Mail/src/inhalt.txt");
3 |
4 | message.setText(mailText);

```

Ergebnis

Man hat ein Programm zum Versenden von E-Mails, dessen Inhalt aus einer Text-Datei gelesen wird. (Umgang mit javamail)

1.4 Aufgabeteil b)

Frage- bzw. Aufgabenstellung

Erweitern Sie ihr Programm so, dass automatisch eine Liste von Empfängern aus der Datei empfaenger.txt eingelesen wird.

Lösung

Man liest die Empfaenger.txt Datei ein und teilt den Inhalt so auf ein String Array auf, sodass hinter einem Index des Arrays immer eine E-Mail steht.

```

1 | ReadFile recipientReader = new ReadFile();
2 | String recipList = recipientReader.readFile("C:/Users/Jan
   |     Augstein/eclipse-workspace/VSPraktikum3Mail/src/empfaenger.txt");
3 | String recipient[] = recipList.split("\n");

```

Diese E-Mails werden dann abgearbeitet und es wird an jede E-Mail-Adresse die E-Mail geschickt.

```

1 | for(int i=0;i<recipient.length;i++) {
2 |     message.setRecipients(Message.RecipientType.TO,
   |         InternetAddress.parse(recipient[i]));
3 |     Transport.send(message);
4 | }

```

Ergebnis

Es werden Empfänger aus einer Text-Datei gelesen und an jede dieser Adressen wird die E-Mail geschickt.

1.5 Aufgabenteil c)

Frage- bzw. Aufgabenstellung

Fügen Sie ihrer Mail einen Anhang hinzu (zum Beispiel ein Bild oder PDF-Dokument).
Erläutern Sie, in welcher Form der Anhang übertragen wird.

Lösung

Zum hinzufügen eines Anhangs wird der Inhalt der E-Mail in 2 Bodys aufgeteilt. Ein Body ist für den Text einer E-Mail.

```
1 | BodyPart msgBodyPart = new MimeBodyPart();
2 | msgBodyPart.setText(mailText);
3 | Multipart multipart = new MimeMultipart();
4 | multipart.addBodyPart(msgBodyPart);
```

Der andere Body wird für den Anhang benutzt.

```
1 | DataSource source = new
2 | | FileDataSource("C:/Users/Jan
3 | | Augstein/eclipse-workspace/VSPraktikum3Mail/src/p03.pdf");
4 | msgBodyPart = new MimeBodyPart();
5 | msgBodyPart.setDataHandler(new DataHandler(source));
6 | msgBodyPart.setFileName("P03.pdf");
7 | multipart.addBodyPart(msgBodyPart);
```

Mit **multipart.addBodyPart** werden einzelne BodyParts einem sogenannten Multi-Part hinzugefügt. Multipart ist ein Objekt, dass einfach aus mehreren BodyParts besteht und somit den Inhalt der E-Mail bildet.

Zum Übertragen vom Anhang wird der **DataHandler** benutzt. DataHandler formt eine Datei (txt, pdf, png) zu einem Byte-Stream um, welcher dann als Anhang übertragen und wieder umgeformt werden kann.

2 Quellen

SSending an Email using the JavaMail API", <http://www.oracle.com/webfolder/technetwork/tutorials/obe/java/javamail/SSendingEmailUsingJavaMail/index.html>, 19.11.2017

mkyong, "JavaMail API – Sending email via Gmail SMTP example", <https://www.mkyong.com/java/javamail-api-sending-email-via-gmail-smtp-example/>, 19.11.2017

"JavaMail API - Authentication", https://www.tutorialspoint.com/javamail_api/javamail_api_authentication.htm, 19.11.2017

srcode, "Java – Mail mit Attachment versenden", <https://srcode.wordpress.com/2011/06/15/java-mail-mit-attachment-versenden/>, 19.11.2017

"Class DataHandler", <https://docs.oracle.com/javase/5/api/javax/activation/DataHandler.html>, 19.11.2017