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

---

21. November 2017

Inhaltsverzeichnis

1 Aufgabe 1

1.1 Frage- bzw. Aufgabenstellung

Eine der bekanntesten Applikationen im Internet ist eMail. Erklären Sie in Stichworten, welche Protolle und Abläufe das Senden und Empfangen von eMails ermöglichen. Wie funktioniert das SMTP Protokoll? Wie funktionieren die Protokolle POP3 und IMAP?
lolol

1.2 Lösung

SMTP

Das SMTP Protokoll funktioniert, indem der MTA (message transfer agent) vom Sender eine TCP-Verbindung mit dem MTA des Empfängers aufbaut. Das eigentliche Senden der Mail funktioniert dann mit der Übertragung von Anfragen in Form von 5 essentiellen Befehlen:

- HELLO <Hostname> : Client übermittelt dem Server seine Identität in Form des FQDN (fully qualified domain name)
- MAIL FROM <Adresse> : Festlegung des Senders
- RCPT TO <Adresse> : Festlegung des Empfängers
- DATA : Einleitung der Übertragung des Nachrichtinhaltes
- QUIT : Beenden der Verbindung

Der MTA des Empfängers kann auf diese Anfragen entsprechend reagieren, entweder nimmt er die Anfragen an und akzeptiert die Übertragung oder er lehnt diese ab.

POP3

Das POP Protokoll wird verwendet um E-Mails mit Hilfe eines Client vom Server abzurufen. Dieses Protokoll benutzt mehrer Befehle um die Mails abzurufen:

- USER : Identifiziert die Mailbox(User)
- PASS : Authentifizierung des Users mit Hilfe eines Passwortes
- QUIT : Beendung der Verbindung zum Server
- STAT : Anforderung der Anzahl der vorhandenen E-Mails
- LIST : Anforderung der Nummer und der Größe aller E-Mails
- RETR : E-Mail vom Server anfordern
- DELE : Angeforderte E-Mail wird gelöscht
- NOOP : Aufrechterhaltung der Verbindung (verhindern von Timeouts)
- RSET : Zurücksetzen der aktuellen Verbindung

Auch hier antwortet der Server entsprechend auf die Anfragen, indem er sie akzeptiert oder sie ablehnt.

IMAP

Das IMAP Protokoll ist eine Art Erweiterung des POP3 Protokolls. Im Gegensatz zum POP3 Protokoll, wartet IMAP nicht auf eine Antwort des Server, sondern sendet direkt mehrere Befehle hintereinander mit einer Kennung. Diese Kennung wird bei der Antwort des Servers mit zurückgeliefert. In der Antwort des Servers steht dann, ob Befehl ein Erfolg oder Misserfolg war und ob der Server noch weitere Informationen erwartet oder ob der Server dem Client die Informationen schickt.

1.3 Ergebnis

Wir kennen die Protokolle, die für den Versand für E-Mials verwendet werden und wissen wie diese funktionieren.

2 Aufgabe 2

2.1 Frage- bzw. Aufgabenstellung

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

2.2 Vorbereitung

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

2.3 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 erzeigen. 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
      javax.mail.Authenticator() {
2     protected PasswordAuthentication
          getPasswordAuthentication() {
3         return new PasswordAuthentication("user",
              "password");
4     }
5 });
```

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 Befehel 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 ensteht 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 gelsen wird. (Umgang mit javamail)

2.4 Aufgabeteil b)

Frage- bzw. Aufgabenstellung

Erweitern Sie ihr Programm so, dass automatich 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.

2.5 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
      FileDataSource("C:/Users/Jan
      Augstein/eclipse-workspace/VSPraktikum3Mail/src/p03.pdf");
2 msgBodyPart = new MimeBodyPart();
3 msgBodyPart.setDataHandler(new DataHandler(source));
4 msgBodyPart.setFileName("P03.pdf");
5 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.

Ergebnis

Der E-Mail kann ein Anhang hinzugefügt werden.

3 Quellen

SSending an Email using the JavaMail API", http://www.oracle.com/webfolder/technetwork/tutorials/obe/j
19.11.2017

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

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

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

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

SSMTP - Simple Mail Transfer Protocol", https://www.elektronik-kompendium.de/sites/net/0903081.htm,
20.11.2017

"POP - Post Office Protocol", https://www.elektronik-kompendium.de/sites/net/0903091.htm,
20.11.2017

ÏMAP - Internet Mail Access Protocol", https://www.elektronik-kompendium.de/sites/net/0903101.htm,
20.11.2017