

TW Mailer Stefan Werner

GITHUB:

<https://github.com/RxndyOG/verteilteSysteme.git>

Path im directory: /twMailer_StefanWerner

https://github.com/RxndyOG/verteilteSysteme/tree/ee67aec009f8fef8cadc19bebead561a50870289/twMailer_StefanWerner

Error Handling wird hier nicht beschrieben werden da es zu viel Platz wegnehmen würde.

1. Login-Prozess:

Der User kann sich, wenn er nicht eingeloggt ist zwischen LOGIN und QUIT entscheiden. QUIT schließt das Programm. LOGIN startet das LOGIN Prozedere.

1. CLIENT sendet LOGIN zu SERVER.
2. SERVER erhält LOGIN und schickt OK oder ERR zurück.
3. CLIENT fragt den User nach uid und password.
4. CLIENT fügt die zusammen und bildet einen INFO string.
 - a. INFO strings werden benutzt um die länge einer Nachricht sowie die Anzahl der Pakete zu zeigen.
 - b. Hier ist der angegebene INFO string gleich der Username und das Password.
 - i. Username\nPassword\n
5. CLIENT sendet INFO string zum SERVER.
6. SERVER parsed INFO string.
7. SERVER schaut mittels ldap ob User Authentik.
8. SERVER sendet OK oder ERR zurück.
9. SERVER speichert User ab und öffnet weitere optionen.
10. CLIENT weiß User ist angemeldet und öffnet andere Funktionen.

2. SEND-Prozess:

Wenn der User angemeldet ist, kann er durch Eingabe von SEND eine Nachricht schicken. Sender wird automatisch eingefügt. User wird nach Subjekt der Nachricht gefragt und nach der Message. SEND-Prozedere startet:

1. CLIENT sendet SEND zu SERVER.
2. SERVER erhält SEND.
3. CLIENT sendet SEND INFO string zu Server.
 - a. INFO string besteht aus TextLength und NumPackages.
4. SERVER schickt OK oder ERR zurück.
5. CLIENT schickt SEND string zum Server.
6. SERVER parsed SEND string.
7. SERVER speichert Nachricht ab.

3. READ-Prozess:

Wenn der User angemeldet ist, kann er mittels READ eine Nachricht lesen. User wird nach der ID der Nachricht gefragt, die gelesen werden soll. READ-Prozedere startet.

1. CLIENT sendet READ zu SERVER.
2. CLIENT sendet READ INFO string.
 - a. INFO string besteht aus TextLength und NumPackages.
3. SERVER schickt OK oder ERR zurück.
4. CLIENT schickt ID zum SERVER.
5. SERVER schickt OK oder ERR zum CLIENT.
6. CLIENT schickt OK oder ERR zurück.
7. SERVER liest Files und schickt die gewollte zum CLIENT.

4. LIST-Prozess:

Wenn der User eingeloggt ist, kann er LIST eingeben und das LIST-Prozedere starten.

1. CLIENT sendet LIST zum SERVER.
2. SERVER schickt OK oder ERR zum CLIENT.
3. CLIENT schickt OK zurück.
4. SERVER schickt Anzahl an Einträge zum CLIENT.
5. SERVER schickt in while schleife alle Nachrichten.
6. CLIENT bekommt die Nachrichten.
7. CLIENT parsed die Nachrichten und printed sie.

5. DEL-Prozess:

Wenn der User angemeldet ist, kann er Nachrichten Löschen. Er kann DEL eingeben, um das DEL-Prozedere zu starten. Er wird nach der ID des zu löschenden Eintrags gefragt.

1. CLIENT sendet DEL zu SERVER.
2. CLIENT sendet DEL INFO string.
3. SERVER schickt OK oder ERR zurück.
4. CLIENT schickt ID zum SERVER.
5. SERVER schickt OK oder ERR zum CLIENT.
6. CLIENT schickt OK oder ERR zurück.
7. SERVER liest Files und löscht den Eintrag mit der Passenden ID.