

Multi-Client TCP socket programmering

DATA2410 - Nettverk- og Skytjenester

Gruppemedlemmer:

s333745 Ana-Maria Poljac

s326325 Maria Teresita Halvorsen

s333783 Camilla Hoelgaard

s333738 Christian Dyrli

Hva har vi gjort?

Vi har laget en Java applikasjon som gjør det mulig å hente ut e-postadresser fra ønsket nettside.

Vi har en klient og en server som kommuniserer ved bruk av Transmission Control Protocol (TCP), som vil si at data blir overført mellom klient og server, uten tap av data.

- Vi har sett på koden til Raju for å skjønne hvordan vi skulle sette opp klient og server. Deretter har vi skrevet koden for å ta imot en string og gjøre den om til en url, for så å returnere alle e-mailene serveren finner på nettsiden.

Hvilke metoder er blitt brukt?

```
public static String messageDecoder(int errorCode)throws IOException{
    String code = "";
    switch (errorCode) {
        case 0: {
            emailExtractor(code); // kaller metoden som henter ut epost, som igjen kaller metoden som formaterer e-posten
            break;
        }

        case 1: {
            code += "!!No email address found on the page!!!";
            break;
        }

        case 2: {
            code+= "!!Server couldn't find the web page!!!";
            break;
        }
    }

    return code;
}
```

- Metode som sender feilmeldinger basert på hva serveren finner.
 - Om den finner epostadresser, blir metoden emailExtractor (forklart lenger ned) kalt på og sender en liste med alle epostaressene funnet på siden fra server til klient.

```

//Metode som tar inn input fra klienten og gjør den om til en URL.
public static URL urlConverter(String clientInput){
    URL url = null;

    try {
        url = new URL( spec: "https://" + clientInput);
    } catch (MalformedURLException e) {
        e.printStackTrace();
    }

    return url;
}

//Åpner en kobling til nettsiden.
public static URLConnection openConnection(URL url) throws IOException {
    URLConnection con = url.openConnection();

    return con;
}

//Tar informasjon som er hentet fra URL og gjør den om til en stream, slik at den kan leses av
//emailExtractor.
public static InputStream getInputStream(URLConnection con) throws IOException {
    InputStream stream = con.getInputStream();
    return stream;
}

```

- **urlConverter(String clientInput)**
 - tar inn en string fra det som skrives i klienten og gjør den om til en URL.
- **openConnection(URL url)**
 - Åpner en kobling til nettsiden.
- **getInputStream(URLConnection con)**
 - Tar inn alt som står på nettsiden slik at den kan leses av emailExtractor metoden.

```

// metode som henter ut eposter vha. regex, returnerer epost
public static ArrayList<String> emailExtractor(String input) throws IOException {

    InputStream stream = inputStream(openConnection(urlReceiver(input)));

    String content = "";
    int i;

    //Legger alle tegn på nettsiden inn i en String
    while((i=stream.read())!=-1){
        try {
            content += (char)i;
        }
        catch(Exception e){
            System.out.println("DET SKJEDDE NOE FEIL");
        }
    }

    ArrayList<String> containedEmails = new ArrayList<>();

    // Regex for å finne epost
    Pattern regex = Pattern.compile(
        regex: "[A-Z0-9._%+-]+@[A-Z0-9.-]+\.\.[A-Z]{2,6}",
        flags: Pattern.CASE_INSENSITIVE | Pattern.MULTILINE | Pattern.DOTALL);

    //Legger epost inn i en liste, dersom de ikke er der fra før
    while(emailMatcher.find()){
        String enEmail = content.substring(emailMatcher.start( group: 0),emailMatcher.end( group: 0));
        if(!containedEmails.contains(enEmail)){
            containedEmails.add(enEmail);
        }
    }

    return containedEmails;
}

```

- **emailExtractor(String input)**
 - metoden som tar inn en stream fra URL.
 - vi har opprettet en regex som skal sjekke om det finnes noen eposter.
 - legger de alle inn i et array.

```
//Metode som formaterer måten epost skal bli skrevet ut på.
public static String emailPrinter(String receivedText) throws IOException {
    String emails = "";

    if(receivedText != null) {
        String [] split = receivedText.split(regex: ",");

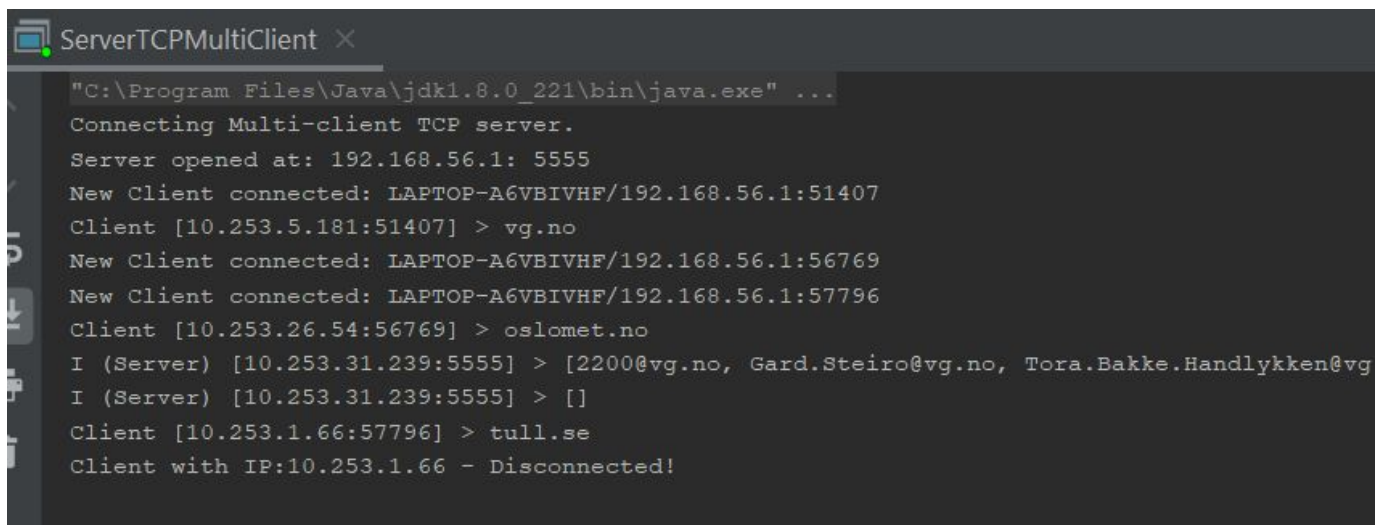
        for (int i = 0; i < split.length; i++) {

            if (i == split.length - 1) {
                emails += split[i];
            } else {
                emails += split[i] + " \n";
            }
        }
    }
    else {
        emails += messageDecoder( errorCode: 2);
    }
    return emails;
}
```

- **emailPrinter**
 - En formaterings metode for alle epostadresser, slik at vi får de printet ut linje for linje.

Hvilke meldinger blir skrevet ut?

Bilde fra server siden:



```
ServerTCPMultiClient X
"C:\Program Files\Java\jdk1.8.0_221\bin\java.exe" ...
Connecting Multi-client TCP server.
Server opened at: 192.168.56.1: 5555
New Client connected: LAPTOP-A6VBIVHF/192.168.56.1:51407
Client [10.253.5.181:51407] > vg.no
New Client connected: LAPTOP-A6VBIVHF/192.168.56.1:56769
New Client connected: LAPTOP-A6VBIVHF/192.168.56.1:57796
Client [10.253.26.54:56769] > oslomet.no
I (Server) [10.253.31.239:5555] > [2200@vg.no, Gard.Steiro@vg.no, Tora.Bakke.Handlykken@vg
I (Server) [10.253.31.239:5555] > []
Client [10.253.1.66:57796] > tull.se
Client with IP:10.253.1.66 - Disconnected!
```

Tilfelle 1: Finner epostadresser på siden

```
Run: ClientTCP x
/Library/Java/JavaVirtualMachines/jdk1.8.0_221.jdk/Contents/Home/bin/java ...
Connecting TCP client!
I (Client) [Christians-MacBook-Pro.local/10.253.5.181:51407] > vg.no
Server [10.253.31.239:5555] > [2200@vg.no
Gard.Steiro@vg.no
Tora.Bakke.Handlykken@vg.no
oyvindb@vg.no
Jane.Throndsen@vg.no
Hanne.Skartveit@vg.no]
I (Client) [10.253.5.181:51407] >
```

Tilfelle 2: Fant ingen epostadresser på siden

```
ClientTCP x
/Library/Java/JavaVirtualMachines/jdk1.8.0_221.jdk/Contents/Home/bin/java .
Connecting TCP client!
I (Client) [Camillas-MacBook-Pro-3.local/10.253.26.54:56769] > oslomet.no
Server [10.253.31.239:5555] > !!No email address found on the page!!!
I (Client) [10.253.26.54:56769] >
```

Tilfelle 3: Websiden eksisterer ikke

```
ClientTCP x
/Library/Java/JavaVirtualMachines/jdk1.8.0_221.jdk/Contents/Home/bin/java ...
Connecting TCP client!
I (Client) [Maria-MacBook-Pro.local/10.253.1.66:57796] > tull.se
!!Server couldn't find the web page!!!
Restart your client
```

Hva gjør applikasjonen?

- Starter serveren, muliggjør for klienter å koble seg til
- Klienten(e) kobler opp mot server vha ClientService klassen som støtter flere klienter samtidig
 - klienten har kjennskap til IP-adressen og portnummeret til server
- Server vet hvilken klient som er koblet opp basert på socketen med unikt portnummer
- Klienten(e) sender en forespørsel (kobling til en nettside) til server
- Server prosesserer forespørselen (bl.a. vha. metoden emailExtractor), og sender tilbake epostadresser som den finner på nettsiden (eller feilmelding om noe går galt).
 - Dersom serveren ikke klarer å koble opp til en side (fordi den ikke eksisterer) vil den sende NULL til klienten og kobler klienten av serveren - samt gi klienten beskjed om dette (code 2).
 - Serveren får beskjed om at klienten(e) har koblet fra.
 - Tilfellet der epost ikke blir funnet vil klienten(e) få svar fra server, og kan deretter sende en ny forespørsel, eller avslutte.

Applikasjonen gjør ovennevnte uten tap av data, slik en TCP protokoll skal.