

## Opgave: TCP server i C#

### Mission:

At designe og implementere et program der kan være en tcp-server med en ekko-protokol.

Der er ikke noget sjov og ballade, bare ganske alm. Console-applikation altså **INGEN RAZOR PAGE**-applikation.

### Background:

CN kap 2.7 p. 189-195

Til programmeringen i server i en C# **Console Application**

- (Server site) TcpListener: [https://msdn.microsoft.com/en-us/library/system.net.sockets.tcpllistener\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.net.sockets.tcpllistener(v=vs.110).aspx)
- (socket (*client site*)) TcpClient: [https://msdn.microsoft.com/en-us/library/system.net.sockets.tcpclient\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.net.sockets.tcpclient(v=vs.110).aspx)

### Opgave A: En Ekko Server

1. In Visual Studio opret en ny solution '**Console App (.Net X)**', og giv det et navn f.eks. 'EchoServer'.
2. Opret en klasse 'Server' og lav en metode 'Start'.
3. I start metoden:
  1. erklær og lav et objekt af TcpListener med parametrene: IP = IPAddress.Loopback og PORT = 7 eller 7777 (port 7 er standard portnummer for en echo server (see [RFC 862](#))).
  2. Start TcpListener'en.
  3. Modtag kald fra klient og behandle forespørgelsen
    1. På TcpListener objektet kald metoden 'AcceptTcpClient()' og gem socketen i en variable fx: 'socket'.
    2. Opret de to kommunikation kanaler til at læse (reading) og skrive (writing) dvs. 'new StreamReader(socket.GetStream())' & 'new StreamWriter(socket.GetStream())'.
  4. Læs en linje fra din stream-reader ('ReadLine()') og gem det i en variable af typen String og navngiv det 'line'.
  5. Skriv linjen tilbage med din stream-writer ('WriteLine(line)') – Husk at 'Flush()' din stream-writer.
4. Så er det bare i program.cs at instantiere et objekt af din Server-klasse og kalde metoden 'Start()'.
5. Kør din server og prøv den med **Socket-Test** programmet

## Opgave B: Refaktor Din Server

1. Du skal refaktorere din server så al kode der behandler en enkelt klient bliver ekstraheret i en metode 'DoOneClient'.
2. Signaturen på metoden kunne være:

```
public void DoOneClient(TcpClient socket)
```

3. Al koden med StreamReader, StreamWriter samt ReadLine, WriteLine flyttes ind i denne metode.
4. Kør Serveren igen og prøv om den stadig virker vha. SocketTest.

## Opgave C: Refaktor Din Server igen

1. Refaktorere din server til at tælle antallet af ord der sendes til serveren og svar tilbage med et tal der viser antallet.
2. Prøv denne nye feature.

## Ekstra A: Overvej hvorledes du kan teste din server

1. Overvej hvordan du tester din server kode.
2. Lav et unit test projekt og test serveren.