

UDP-programmering med Python

Formål: At lave det første UDP Socket program i Python (modtager hhv. afsender)

Baggrund:

- Hav 'SocketTest' installeret (se [tools](#))
- Hav 'PyCharm' installeret (se installations guide)
- Læs i "Computer Networking" bogen kap 2.7: ss.184-189

Opgave 1.1: Simple Echo Modtager (server)

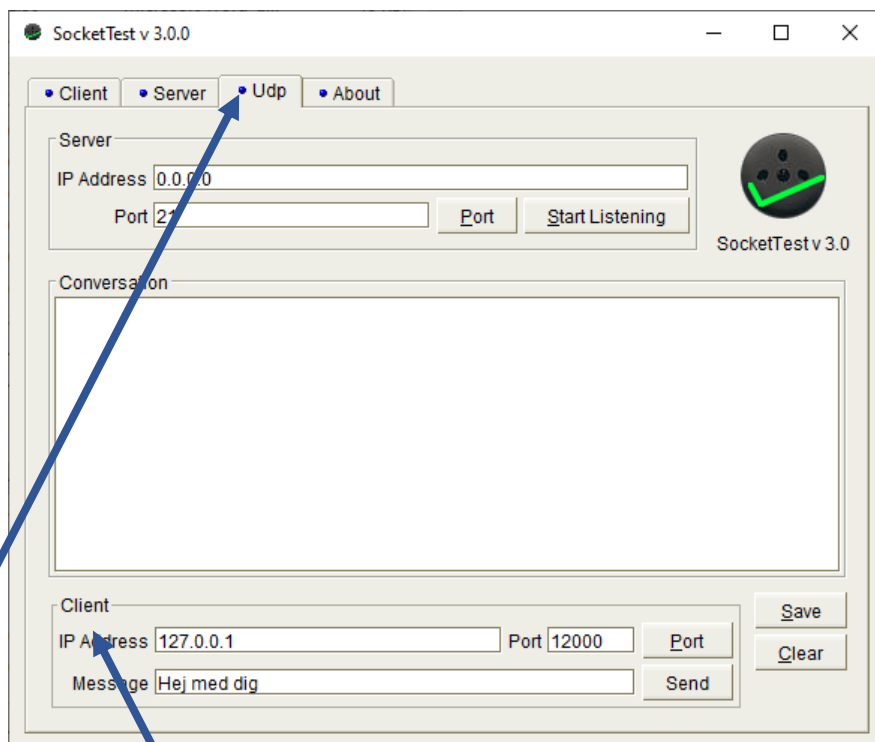
Du skal lave en simpel ekko-modtager (se [RFC862](#) for yderligere information)

I bogen "Computer Networking" kap 2.7.1: ss.188 er et simpelt ekko program beskrevet.

Lav et nyt python-projekt og slet det der står i main.py. Tag og indskriv (kopier) det der står i bogen s 188

Opgave 1.2: Test din Echo Modtager

Du skal nu benytte SocketTest til at test din Simple Echo Modtager. Start SocketTest – og du får:



Vælg UDP-fanen og skriv port + message i client-delen – tryk send

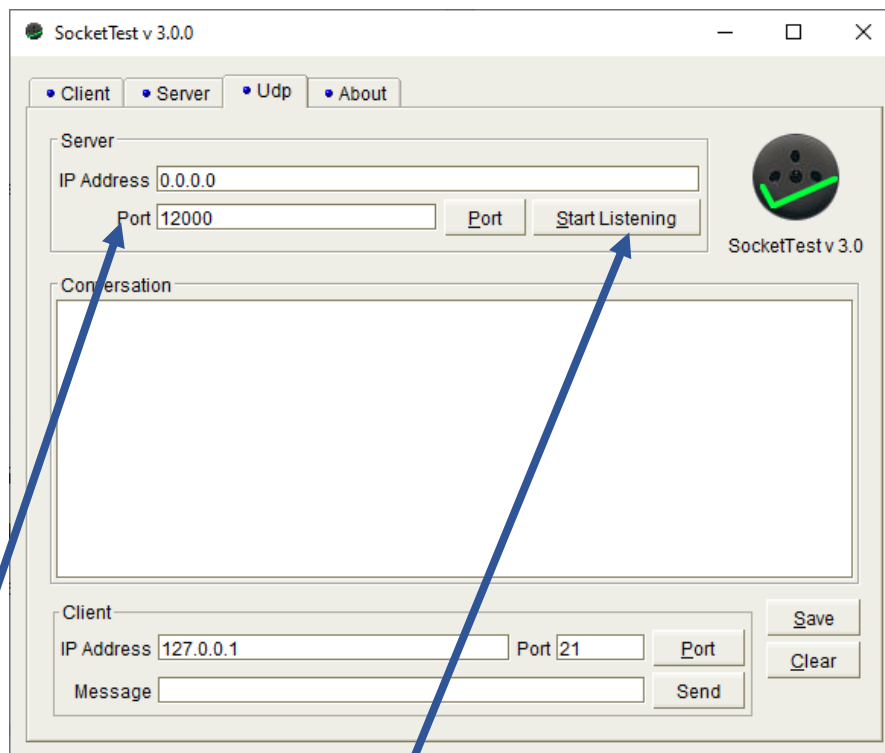
Opgave 2.1: Simple Echo Afsender

Du skal nu lave en UDP Socket Afsender som laver et opkald til din Simple Echo Modtager.

Lav et nyt Python projekt og slet det der står i main.py. Tag og indskriv (kopier) det der står i bogen s 186. Du skal ændre Servername = 'hostname' til 'localhost'

Opgave 2.2: Test Simple Echo Client

Du skal nu benytte SocketTest programmet som en UDP-Modtager. Husk at klikke på UDP-fanen.



Indsæt portnummeret 12000 og klik 'Start Listening' – i server-delen. Nu vil SocketTest fungere som en modtager.

Start dit client program og se der kommer en forespørgsel til SocketTest-modtageren – du kan sende en svar ved at skrive en 'message' og klikke 'send'.

Opgave 2.3: Prøv din Client mod din Server

Du skal her prøve begge dine programmer.

Start din afsender (uden at starte din modtager)– hvad sker der?

Start din modtager og derefter start din afsender – hvad sker der nu ?

Kan du starte en UDP-modtager på port 12000 samtidig med en TCP-server på 12000?

Opgave 3: Udvid (refactor) din UDP-modtager

Udvid din UDP-modtager (server) til at være concurrent, dvs. håndtere de enkelte forespørgsler i hver sin tråd (thread).

Prøv om det virker.