# Dokumentace projektu: Alfa 4 (P2P Chat)

* Autor: Šimon Bernard C4b
* Mail: [bernard@spsejecna.cz](mailto:bernard@spsejecna.cz)
* Datum: 25. 02. 2024
* Název školy: Střední průmyslová škola elektrotechnická, Ječná 30, Praha, Česká republika
* Typ projektu: Alfa 4
* Zaměření projektu: Peer to peer chat
* Vývojové prostředí: PyCharm

## Použité externí knihovny

Flask => <https://flask.palletsprojects.com/en/3.0.x/installation/#install-flask>

* Path/to/project: pip install Flask

## Chod aplikace

* Funkčnost programu můžeme vidět např. v příkazovém řádku, kde se připojíme na ssh pomocí příkazu: ssh -p 20515 [jouda@dev.spsejecna.net](mailto:jouda@dev.spsejecna.net) a zadáme heslo: (na vyžádání přes email). Následně zadáme příkaz: sudo systemctl start chat, poté si můžeme otevřít ještě jeden příkazový řádek a zadat další ssh příkaz: ssh -p 20110 [jouda@dev.spsejecna.net](mailto:jouda@dev.spsejecna.net) a zadat heslo: jooouda. Následně zadejte příkaz: journalctl -fu chat a uvidíte chod a funkčnost programu.

### Spuštění aplikace:

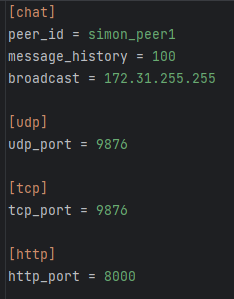
* 1. Příkazový řádek:
  + Zadejte: ssh -p 20515 [jouda@dev.spsejecna.net](mailto:jouda@dev.spsejecna.net)
  + Heslo: (heslo)
  + cd /home/chat
  + /home/chat: sudo pip install Flask
  + /home/chat: sudo systemctl start chat (spuštění programu)
* 2. Příkazový řádek
  + Zadejte: ssh příkaz: ssh -p 20110 [jouda@dev.spsejecna.net](mailto:jouda@dev.spsejecna.net)
  + Heslo: jooouda
  + journalctl -fu chat (zobrazí chod programu)
* 1. Příkazový řádek
  + /home/chat: sudo systemctl stop chat (ukončení programu)

### Popis:

* Jedná se o peer to peer chat, kde moje peer komunikuje v síti s ostatními peerami a vzniká tak p2p chat, který historii zpráv ukládá do textového souboru messages.txt a je konfigurovatelný.
* UDP
  + Program každých 5 sekund pošle do sítě UDP broadcast na port 9876 (hello message) a čeká na zpětnou odpověď od ostatních peerů (ok message)
* TCP
  + Pokaždé co program objeví novou peeru naváže s ní trvalé spojení na portu 9876 a komunikace funguje na podobné bázi jako UDP pomocí JSON
* Webové API
  + url/messages => by mělo vypsat všechny zprávy
  + můj program zprávy ukládá do souboru messages.txt, a tak si zprávy můžete vypsat pomocí příkazu cat messages.txt

## Konfigurace

* Konfigurace probíhá ve složce /conf/configuration.ini



# Uložení zpráv

* V TCP se po úspěšném handshaku volá metoda save\_messages(), která ukládá tyto data (historie zpráv) do souboru messages.txt
* Aktuálnost těchto zpráv se zajišťuje tím, že se data v souboru přepisují

## Chybové stavy

* udp\_server:
  + ValueError in udp server
  + Exception in udp sever
* udp\_client:
  + Exception in udp klient
* tcp\_handler:
  + Eception in tcphandler
  + Error saving data to file
* app:
  + File not found at messages.txt
  + Error fading data from file

## Závěrečné shrnutí

* Projekt Alfa 4 hodnotím neutrálně. Propojení práce mezi předměty je zajímavé, ale značně mi nevyhovovalo testování, které bylo přímo závislé i na pracích ostatních studentů, tudíž například chyba v programu někoho jiného ovlivnila testování jiných prací. I přes to byla práce zajímavá, a jak jsem již zmínil, propojení předmětů mě zaujalo.