

Micro-messenger

Le but est de créer une petite application de messagerie instantanée.

Tout fonctionnera en ligne de commande.

Il y a deux modes de fonctionnement :

- Le mode client, pour initier une conversation.
- Le mode serveur, pour se mettre en attente d'une nouvelle conversation.

Mode client

En tant que client, on fournit l'hôte (adresse IP ou nom d'hôte qualifié) et le port du serveur.

Lorsqu'on est connecté, on doit recevoir les messages des autres utilisateurs.

On doit pouvoir envoyer des messages, ligne par ligne.

Mode serveur

En tant que serveur, on fournit l'adresse IP et le port d'écoute.

Le programme se met en attente d'une connexion de la part d'un client.

Pour chaque client connecté, un message est affiché avec l'adresse IP et le port source.

Tous les messages reçus doivent également s'afficher, comme pour un client classique.

Le serveur n'émet pas de messages hormis une bannière de bienvenue.

Format des messages

Chaque message s'affichant en console doit être au format :

```
<hours>:<seconds> <AM | PM> <username> : <message>
```

hours: l'heure actuelle seconds: la seconde actuelle username: Le nom de l'utilisateur message: Le message

CLI

Mode client :

```
micro-messenger -c -h <host> -p <port> -u <username>
```

- -c: activer le mode de connexion (obligatoire)

- **-h**: le nom d'hôte ou adresse IP du serveur (obligatoire)
- **-p**: le port du serveur (optionnel, par défaut 19337)
- **-u**: un nom d'utilisateur qu'on choisit (optionnel, valeur par défaut est anonymous).

Mode serveur :

```
micro-messenger -l -a <addr> -p <port> -b <banner>
```

- **-l**: activer le mode d'écoute (obligatoire)
- **-a**: l'adresse d'écoute (optionnel, par défaut 0.0.0.0)
- **-p**: le port d'écoute (optionnel, par défaut 19337)
- **-b**: une bannière à afficher lors de la connexion (optionnel)

Contraintes techniques

- Tout doit être fait en Java
- Aucune librairie ne doit être utilisée.
- Tout doit être en console, pas d'interface graphique nécessaire.
- Aucun service externe ne doit être utilisé.
- Interdiction d'utiliser un protocole applicatif existant (http, ssh...), on utilisera plutôt des sockets purs !