
SAÉ 3.03 Système et Réseau
Réseau social Tuit'o

1 Sujet

Le but de cette SAÉ est de développer une application client-serveur d'un réseau social : les utilisateurs peuvent publier des messages, et suivre d'autres utilisateurs pour consulter leurs publications.

Il y a deux aspects à cette application :

- le client qui doit se connecter au serveur pour afficher les messages des personnes qu'il suit et poster ses propres publications,
- le serveur qui reçoit les messages, et les réexpédie aux bons utilisateurs.

1.1 Client

Le client se connecte au serveur en précisant l'adresse IP de celui-ci (ou son nom), ainsi que son nom d'utilisateur. Il voit ensuite la liste des différents messages postés par les utilisateurs auxquels il est abonné, dans l'ordre chronologique (une limite du nombre de messages affichés devra être implantée).

Si le nom d'utilisateur n'est pas enregistré sur le serveur, le client en sera informé. On lui proposera de créer ce compte, et de commencer le suivi d'autres utilisateurs.

Sur chaque message, on peut voir en plus du contenu, le nom de l'expéditeur, la date/heure, le nombre de likes et l'identifiant du post.

L'utilisateur peut poster un message, celui-ci sera envoyé à tous ses abonnés, sans que ceux-ci n'effectuent la moindre action (pas d'actualisation).

L'utilisateur peut également effectuer plusieurs actions. Ces actions sont précédées de "/" Plusieurs commandes doivent être disponibles :

- `/follow <nom_utilisateur>` permet de s'abonner à un nouvel utilisateur. Si celui-ci n'existe pas un message d'erreur s'affiche
- `/unfollow <nom_utilisateur>` se désabonner
- `/like <id_message>` : aime un message
- `/delete <id_message>` : supprime un de ses messages

Bien évidemment, ces commandes envoyées au serveur, ne devront pas être envoyées aux abonnés. Cette application client est utilisable dans un terminal.

1.2 Serveur

Le serveur doit accepter des connexions de la part de nouveaux clients. Lorsqu'il reçoit un message, il doit le réenvoyer aux bons utilisateurs. Le serveur reçoit bien évidemment plusieurs messages en même temps, et doit donc être capable de les traiter en parallèle.

Des commandes particulières sont accessibles en lignes de commande sur le serveur

- `/delete <id_message>` : supprime un message
- `/remove <nom_utilisateur>` : supprime un utilisateur, ainsi que tous ses messages

Le serveur est accessible en ligne de commande.

Le format des messages échangés doit être json, afin de garantir la compatibilité des clients et des serveurs.

```
message:
{"id" : 987697
"user" : "toto",
"content" : "Hello world !",
"date" : "2024-01-20T09:00:00Z",
"likes" : 3}
```

1.3 Extensions

1.3.1 Persistance

Faites en sorte que le serveur soit persistant. Si celui-ci redémarre, il doit revenir dans l'état dans lequel il était.

De la même manière, vous pouvez faire un client qui se souvient du serveur, et du nom de l'utilisateur.

1.3.2 Interface graphique

Développer une interface graphique pour le client

2 Déroulement de la SAÉ

La SAÉ se fait par groupe de deux. Elle se déroule du 04/12 au 20/01. Le développement se fait en Java.

Rendu intermédiaire le 22/12/2023 :

- Diagramme de classes
- Une archive avec un prototype de l'application : un client et un serveur qui peuvent communiquer avec envois de messages simples (pas de commandes)

Rendu final Le **20/01/2024** sur Célène :

- Diagramme des classes final
- Application fonctionnelle
- Rapport : Manuel d'utilisation et justification des choix techniques
- documentation (javadoc)

Résumé	Développer une application communiquant sur le réseau via des sockets, et utilisant des threads
Compétences Visées	<ul style="list-style-type: none">— Compétence 3 : Administrer des systèmes informatiques communicants complexes— Compétence 6 : Travailler dans une équipe informatique
Heures SAÉ	6 heures entre le 04/12 et le 20/01
Liste des ressources mobilisées	<ul style="list-style-type: none">— R3.05 : Programmation Système— R3.06 : Architecture des réseaux
Livrables sur Célène	<ul style="list-style-type: none">— Code source commenté— Diagramme de classes— Rapport et Manuel utilisateur— Documentation Javadoc