

Rapport SAE 3.03

Système et réseaux

Par Ludmann Dorian, Brion Adèle et Dubois Tom

| | |
|--|----------|
| I - Manuel utilisateur | 2 |
| 1 - Connexion | 2 |
| 2 - Côté serveur | 2 |
| 3 - Côté client | 2 |
| II - Explication des choix techniques | 4 |

I - Manuel utilisateur

1 - Connexion

Pour mettre en place la liaison entre la BD et le serveur, veuillez suivre les instructions disponibles dans le fichier "configBD.txt".

2 - Côté serveur

```
----- Command -----  
| Enter /deleteuser {username} to delete an user |  
| Enter /deletemsg {id} to delete a TUIT |  
| Enter /deleteall to delete everythings |  
-----
```

/deleteuser:

- Permet de supprimer un utilisateur, ces identifiants, mot de passe et TUIT

/deletemsg:

- Permet de supprimer un TUIT en indiquant son id

/deleteall:

- Supprimer l'ensemble des données stockés dans la BD (notamment utile lors de séances de tests)

3 - Côté client

```
--- Connexion au serveur ---  
Souhaitez vous vous connecter ou vous inscrire ?  
1. Connexion  
2. Inscription  
3. Quitter
```

Taper 1:

```
--- Connexion ---  
Identifiant : Tom  
Mot de passe : 1234
```

Demande les identifiants et mot de passe, si c'est correct, vous connecte à l'application, sinon vous renvoie au menu d'accueil.

Taper 2:

```
--- Inscription ---  
Identifiant : Kyo  
Mot de passe : 1234  
  
--- Inscription réussie ---  
Vous êtes connecté en tant que Kyo.
```

Renseignez un identifiant et un mot de passe pour créer le compte. Vous serez automatiquement connecté

Taper 3:

Quitte l'application

Une fois connecté....

Vous serez notifié des derniers messages envoyés par les utilisateurs que vous followez, et vous aurez ensuite la possibilité d'interagir avec le terminal. Pour voir en détail les commandes possibles, il vous suffit de taper “ /help”. Ensuite, tout est défini.

Tout autre message ne commençant pas par un “/” sera envoyé sous forme de message. Toute personne qui vous follow va recevoir ce message.

```
--- Bienvenue sur Tuit'O, voici ce que vous avez manqué! ---  
  
Enter a message to send ('/quit' to exit or '/help' to see the available commands):  
-help  
You sent: -help  
/help  
  
----- Command -----  
|                               |  
|       Enter a message to send       |  
|       Enter /follow {username} to follow someone       |  
|       Enter /unfollow {username} to unfollow someone       |  
|       Enter /followers to see who follow you       |  
|       Enter /following to see who you follow       |  
|       Enter /like {id} to like a TUIT       |  
|       Enter /unlike {id} to unlike a TUIT       |  
|       Enter /getlikes {id} or /getnblikes {id} or /nblikes {id} to see the likes of a TUIT       |  
|       Enter /history {x} or /historique {x} to see your x last TUIT       |  
|       Enter /remove {id} or /delete {id} to delete one of your TUIT       |  
|       Enter /followers to see your followers       |  
|       Enter /quit to exit       |  
|                               |  
|-----Details-----|  
| - {username} refer to a pseudo       |  
| - {id} refer to a TUIT id       |  
|-----|
```

II - Explication des choix techniques

Nous avons décidé d'implémenter ce problème de client/serveur d'une manière que le client n'ait pas besoin de stocker temporairement d'information, ni d'avoir besoin d'accéder à la BD. En effet, c'est le serveur, et uniquement lui qui peut accéder aux informations stockés dans la base de données, afin de sécuriser le tout.

Afin de rendre l'application plus ergonomique, nous avons doté le client et le serveur d'un thread, afin de partager la tâche lecture des sockets et écriture dans le terminal.

L'utilisation de la BD permet d'éviter les conflits, ce qui rend la tâche bien plus facile et efficace. L'utilisation d'un client handler était alors favorisé, puisqu'en plus de permettre au serveur de gérer plusieurs clients, le fait que les conflits étaient gérés par la BD permet alors à l'ensemble des client handlers d'appeler des fonctions reliées aux données facilement sans avoir besoin de prendre beaucoup de précautions.

La création d'un fichier de configuration de la BD a été proposée afin de ne pas avoir besoin de recompiler le code lorsqu'une autre BD a dû être utilisée, en plus d'éviter de "hardcode" la connexion.

Sachant que nous n'avons pas intégré d'interface graphique, nous n'avons pas pour autant négligé l'aspect ergonomique et propre de l'application. Nous avons rendu le rapport au terminal en une expérience positive pour l'utilisateur, en le guidant, tout en lui donnant un maximum de feedback quant à ses actions.

Par Ludmann Dorian (2A1), Brion Adèle (2A2) et Dubois Tom (2A2)