

Implémentation du Client Matou

Le Client Matou repose sur 3 classes :

- ClientCore : C'est la partie "haut niveau" du client. Cette partie est responsable de l'implémentation TCP en mode "concurrency" du client.
- ClientSession : Il y a une seule instance de cet objet par exécution. Elle contient les informations relatives à l'exécution en cours ainsi que les SocketChannel ouvertes.
- ClientEvent : Interface permettant de généraliser les événements demandés par l'utilisateur (envoyer un nouveau message, envoyer une demande de connexion privée...).

Le client démarre 1 thread général pour l'envoi des messages, 1 thread pour la réception des messages publics, 1 thread par client pour la réception des messages privés et 1 thread pour nettoyer les connexions inactives.

D'autres threads intermédiaires sont parfois créés pour une tâche spécifique (par exemple : l'envoi et la réception de fichiers).

L'interface utilisateur par défaut est une interface shell. Les événements sont créés à partir des commandes suivantes :

- */open pseudo*
Demander à ouvrir une connexion privée avec *pseudo*. Il n'est pas possible d'envoyer plusieurs demandes à un même client.
- */accept pseudo*
Accepter la demande de connexion privée de *pseudo*. Le client cible doit avoir demandé la connexion privée.
- */pv pseudo message*
Envoyer un message privé à *pseudo*.
- */file pseudo chemin*
Envoyer un fichier à *pseudo*.
- */close pseudo*
Fermer la connexion privée avec *pseudo*.

La saisie d'un pseudo est insensible à la casse.

Les fichiers reçus sont enregistrés dans le répertoire *./files*.

Le nom du fichier reçu est généré aléatoirement et précédé du pseudo du client ayant envoyé le fichier.

L'envoi et la réception des fichiers ne bloquent pas l'envoi et la réception de messages.

Les fichiers sont envoyés et reçus par tronçon de 4 Kio. La taille du fichier étant encodée sur un Long, il est possible d'échanger des fichiers très volumineux.

Le nettoyage des connexions inactives n'est pour le moment implémenté que pour surveiller la connexion publique avec le serveur.

Une connexion est jugée inactive si le client est bloqué depuis plus de 3 secondes sur la lecture des arguments d'une requête (à l'exception du 1er argument qui correspond au type de requête).