

TP Réseau : Chat

Introduction

Ce TP a pour objectif la création d'un chat en ligne par le biais du développement conjoint d'application Java côté serveur et client. Pour ce faire, nous avons dans un premier temps implémenté un chat utilisant la technologie RMI : Remote Method Invocation. Puis, nous avons dans un second temps travaillé sur l'implémentation d'un second chat utilisant le systèmes de sockets pour effectuer des échanges entre un client et un serveur.

Chat RMI

Tous d'abord, nous avons développé une application côté serveur fonctionnelle qui initialise le registre et « bind » un Session dans celui-ci. Cet objet Session est donc un objet remote qui va pouvoir être utilisé par tout client pour récupérer l'état du chat (chat persistant) mais également interagir avec le serveur et modifié cet état. Pour cela, nous avons implémenté une application côté client avec interface graphique. Grâce à cette interface, l'utilisateur va pouvoir faire appel aux méthodes inscrites dans l'interface de l'objet Session, récupéré à la connexion. Les méthodes appelées vont dépendre du type de commandes entrées par l'utilisateur. Le protocole utilisé est le suivant :

Commandes	Significations
<i>/connect username</i>	L'utilisateur se connecte. <i>Username</i> est son pseudonyme.
<i>message</i>	L'utilisateur envoie un message que tous les utilisateurs connectés sur le chat verront.
<i>/private username message</i>	L'utilisateur envoie un message privé à un utilisateur ayant comme pseudonyme <i>username</i> .
<i>/disconnect</i>	L'utilisateur se déconnecte.

Chat Sockets

Nous avons ensuite travaillé sur le développement de ce même chat mais en utilisant la technologie des sockets. Lorsqu'un client se connecte au serveur, un socket le caractérisant est créé côté serveur et permet la création d'un objet thread nommé : ClientThread qui va être lancé directement. Ce thread va nous permettre de traiter indépendamment les requêtes de chacun des clients connectés. Ces requêtes sont donc caractérisés par un objet Request qui part son type,options,contenue va pouvoir servir de support d'information entre le serveur et le client. Différents objets request vont donc transiter entre le serveur et le client à travers les

objectInputStream et objectOutputStream liant les sockets clients côté serveur et côté client. L'application côté client est également munie d'une interface graphique grâce à laquelle l'utilisateur peut interagir avec le serveur et par la même occasion le chat et ceci par le biais du même protocole énoncé précédemment. Nous avons également implémenté l'affichage de la liste des utilisateurs connectés sur cette interface pour améliorer l'expérience utilisateur. Cependant, dans ce second chat, nous avons tenté d'implémenter la persistance d'une manière différente mais nous n'avons à l'heure actuelle toujours pas réussi à résoudre le problème rencontré. Les erreurs obtenues sont les suivantes : EOF Exception. L'erreur se situe lorsque que l'on tente de récupérer les objets stockés sur le fichier de logs. On souhaite récupérer requête par requête mais le fichier se remplit mal, peut être une mauvaise gestion du flux d'écriture dans le fichier. Nous avons également quelques problèmes avec l'affichage de la liste des utilisateurs avec la technologie Socket mais nous avons réussi à le régler en rectifiant une erreur que nous avons laissé lors d'un changement de structure effectué durant le développement.

Fonctionnalités non implémentées

Par manque de temps, nous n'avons pas pu implémenter de nombreuses fonctionnalités secondaires auxquelles nous avons pensé lors de l'élaboration des spécifications du produit. Voici une liste non exhaustive de ces fonctionnalités :

- ajout de chat room (divers chats parallèles accessibles par la même application)
- ajout de couleur de textes (ex : message privé en bleu)
- ajout d'onglet (ex : connexion à de multiples chats room, onglet messages privés de tel utilisateur)
- ajout de gestion de paramètres (ex : augmentation de police de texte, couleur, désactivation message privé, ...)
- amélioration de l'interface graphique (ex : personnalisation graphique plus poussée, ...)
- intégrer la possibilité d'écrire en gras/italique à l'aide d'options ou de balises
- ajout d'émoticônes pour l'utilisateur

Voici un résumé des fonctionnalités implémentées sur les deux chats :

- connexion
- déconnexion
- message classique
- message privé
- interface graphique simple
- affichage liste utilisateurs
- persistance (seulement sur le RMI)