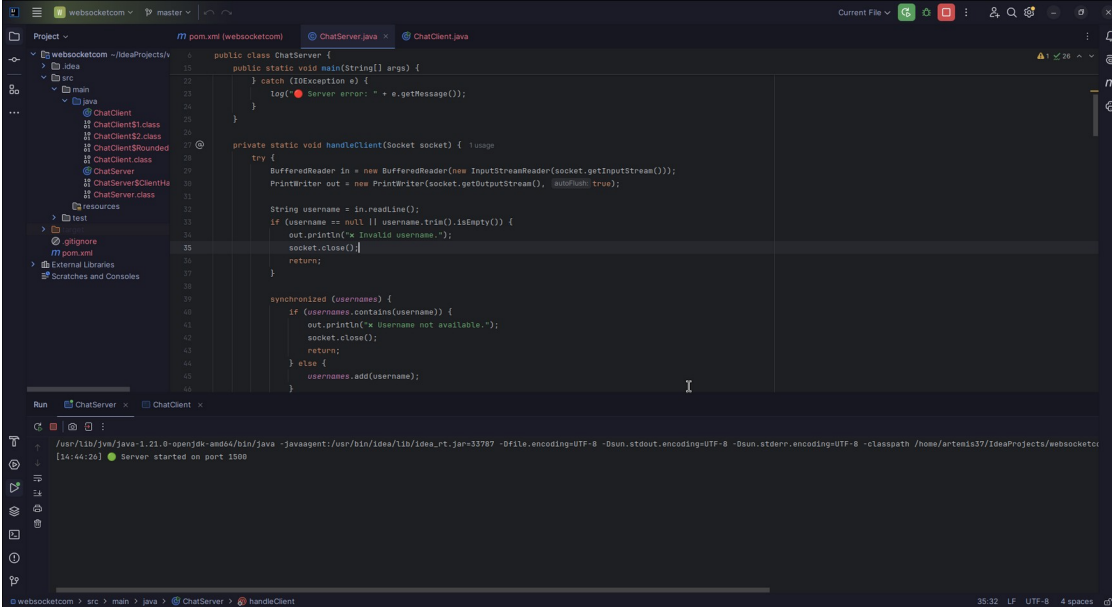


Présentation du Projet : Serveur Multithreadé en Java

Contexte

Ce logiciel est un client de chat modulaire conçu pour offrir une expérience conversationnelle, autonome et extensible.

Il intègre une interface simple de communication, une gestion des groupes par commandes, et une interface néon arrondie pensée pour la clarté émotionnelle.



```
public class ChatServer {
    public static void main(String[] args) {
        try {
            Server server = new Server(1500);
            server.start();
        } catch (IOException e) {
            log("Server error: " + e.getMessage());
        }
    }

    private static void handleClient(Socket socket) {
        try {
            BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
            PrintWriter out = new PrintWriter(socket.getOutputStream(), true);

            String username = in.readLine();
            if (username == null || username.trim().isEmpty()) {
                out.println("Invalid username.");
                socket.close();
                return;
            }

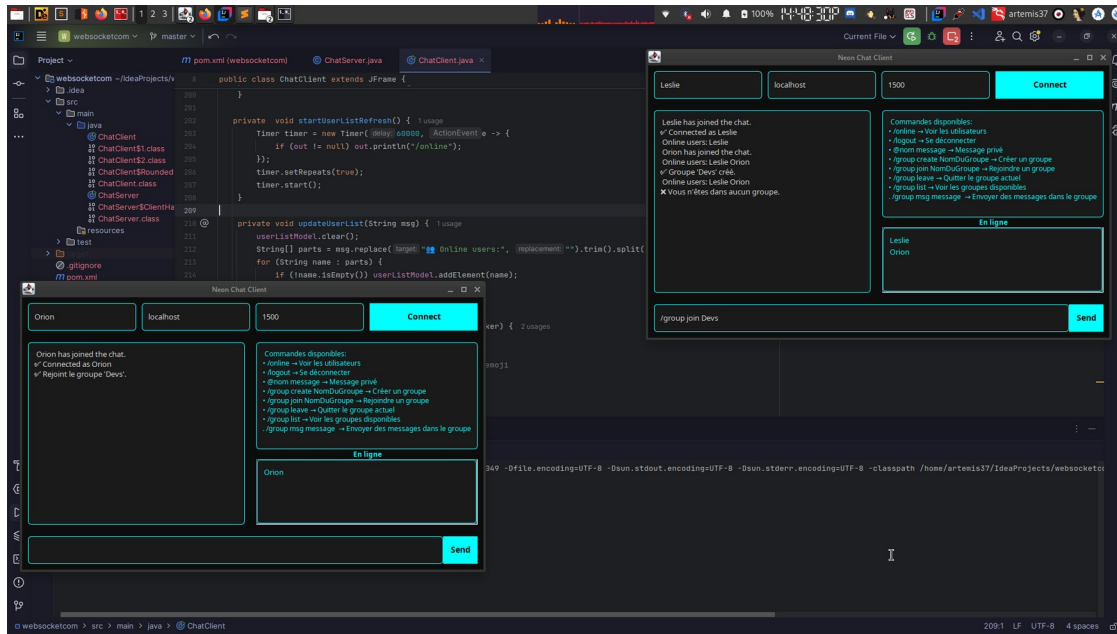
            synchronized (usernames) {
                if (usernames.contains(username)) {
                    out.println("Username not available.");
                    socket.close();
                    return;
                } else {
                    usernames.add(username);
                }
            }
        }
    }
}
```

Run ChatServer ChatClient

/usr/lib/jvm/java-1.21.0-openjdk-amd64/bin/java -javaagent:/usr/bin/idea/lib/idea_rt.jar=33787 -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath /home/artemis37/ideaProjects/websocketc: [14:44:26] Server started on port 1500

Fonctionnalités Principales

- **Gestion des groupes par commandes textuelles**
 - /group create nom_du_groupe
 - /group join nom_du_groupe
 - /group leave nom_du_groupe
- **Assistant conversationnel avec mémoire contextuelle**
- **Interface néon, arrondie, avec feedback en temps réel**
- **Parsing robuste et validation des commandes**
- **Architecture modulaire pour extensions techniques**



⚙ Fonctionnement

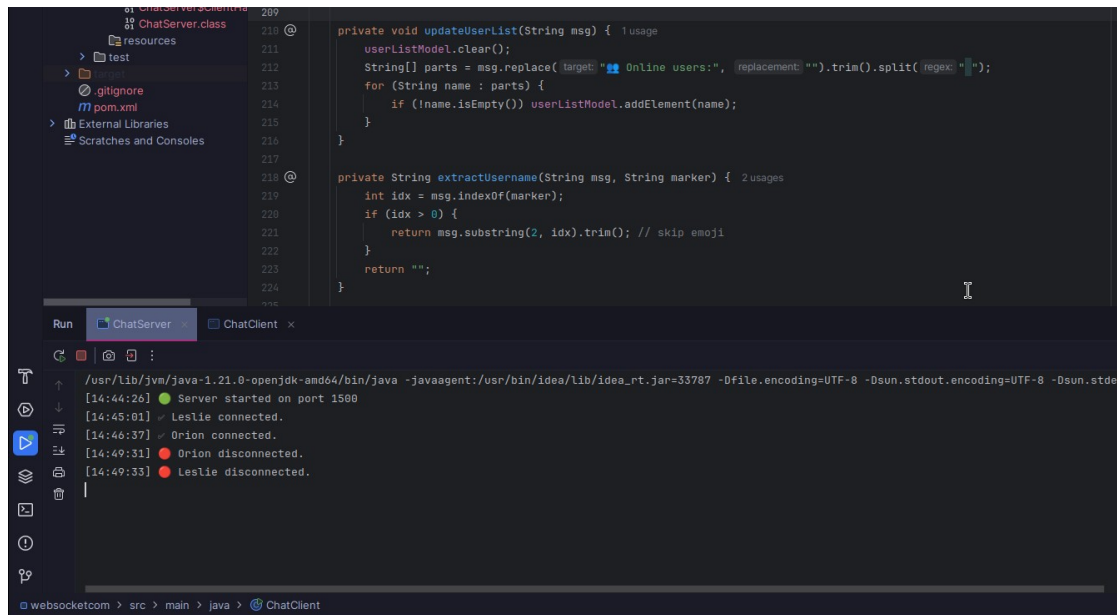
♦ Lancement

L'utilisateur ouvre le client et interagit directement avec l'assistant.
Aucun bouton GUI n'est nécessaire : tout se fait par commandes ou dialogue naturel.

♦ Interaction

Les commandes sont analysées et exécutées avec retour immédiat.
L'assistant guide l'utilisateur en cas d'erreur ou d'ambiguïté.

Gestion des Utilisateurs et des déconnexions



The screenshot shows an IDE with a project named 'websocketcom'. The file explorer on the left shows 'resources', 'test', 'pom.xml', and 'External Libraries'. The main editor displays the 'ChatServer.class' file with the following Java code:

```
209
210 @
211 private void updateUserList(String msg) { 1 usage
212     userListModel.clear();
213     String[] parts = msg.replace( target: "Online users:", replacement: "").trim().split( regex: "\\s");
214     for (String name : parts) {
215         if (!name.isEmpty()) userListModel.addElement(name);
216     }
217
218 @
219 private String extractUsername(String msg, String marker) { 2 usages
220     int idx = msg.indexOf(marker);
221     if (idx > 0) {
222         return msg.substring(2, idx).trim(); // skip emoji
223     }
224     return "";
225 }
```

The Run console at the bottom shows the following output:

```
/usr/lib/jvm/java-1.21.0-openjdk-amd64/bin/java -javaagent:/usr/bin/idea/lib/idea_rt.jar=33787 -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8
[14:44:26] Server started on port 1500
[14:45:01] Leslie connected.
[14:46:37] Orion connected.
[14:49:31] Orion disconnected.
[14:49:33] Leslie disconnected.
```

◆ Feedback

Chaque action déclenche un retour visuel : animations, messages, ou transitions.
Le panneau d'aide s'adapte dynamiquement selon le rôle et le contexte.

◆ Extension

De nouveaux modules peuvent être ajoutés pour : - Etendre la communication sur tout type de support - Automatiser des tâches techniques - Intégrer des modèles ML - Enrichir les interactions avec des styles d'émojis etc.....

Design & Expérience Utilisateur

- Couleurs néon : bleu électrique, violet, vert acide
- Bords arrondis, ombrages doux, transitions fluides
- Interface intuitive, émotionnellement guidée
- Équilibre entre puissance technique et clarté visuelle

Notes Techniques

- Parsing des commandes basé sur découpage explicite
 - Gestion des erreurs avec messages guidés
 - Architecture pensée pour multi-threading et extensibilité
-

Demo et code disponible à l'adresse

https://github.com/Artemisxxx37/multi_threaded_java_server.git