Documentation de l'application de chat Java RMI

Nissane Youssef

Introduction

L'application de chat Java RMI avec GUI est un projet conçu pour faciliter la communication en temps réel entre plusieurs utilisateurs sur un réseau. En combinant la technologie d'invocation de méthode à distance (RMI) de Java avec une interface utilisateur graphique (GUI) élaborée en utilisant Swing, ce projet offre une expérience de discussion transparente. Que ce soit pour un usage personnel ou professionnel, cette application est conçue pour répondre aux exigences de la communication moderne en offrant une plateforme intuitive permettant aux utilisateurs de s'engager dans des conversations sans effort.

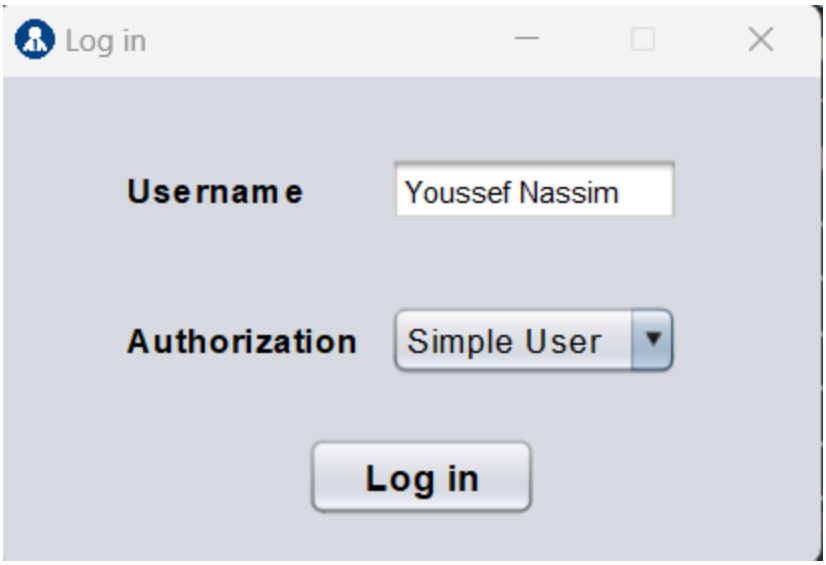
Objectif du projet

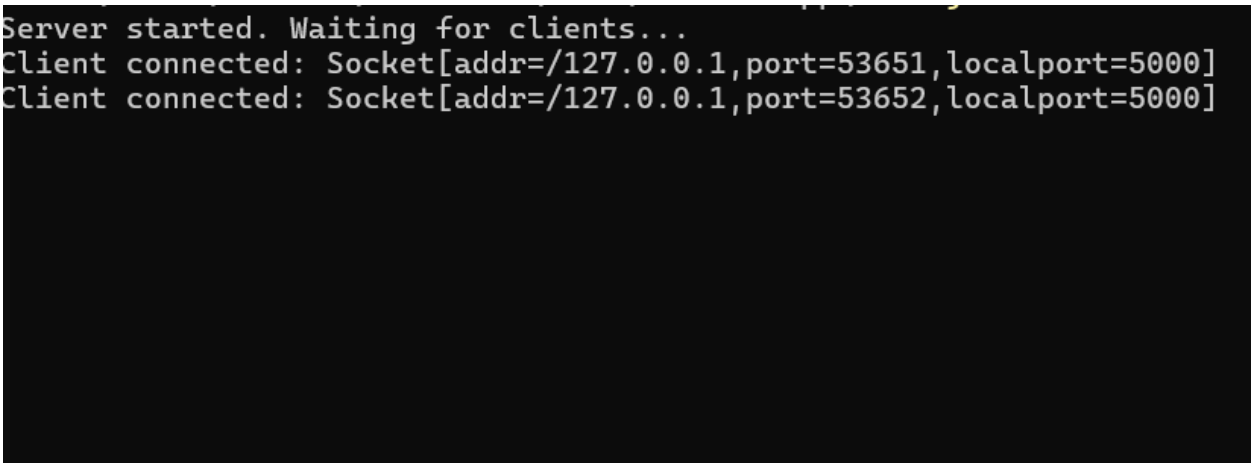
L'objectif principal de l'application de chat Java RMI avec GUI est de répondre au besoin croissant de solutions de communication en temps réel efficaces et fiables. En exploitant la technologie RMI, le projet vise à établir un cadre de communication robuste capable de gérer plusieurs connexions concurrentes tout en garantissant l'intégrité et la sécurité des données. L'objectif principal est de créer une application de chat évolutive offrant une expérience utilisateur riche grâce à son interface graphique, permettant aux utilisateurs de se connecter, de communiquer et de collaborer facilement.

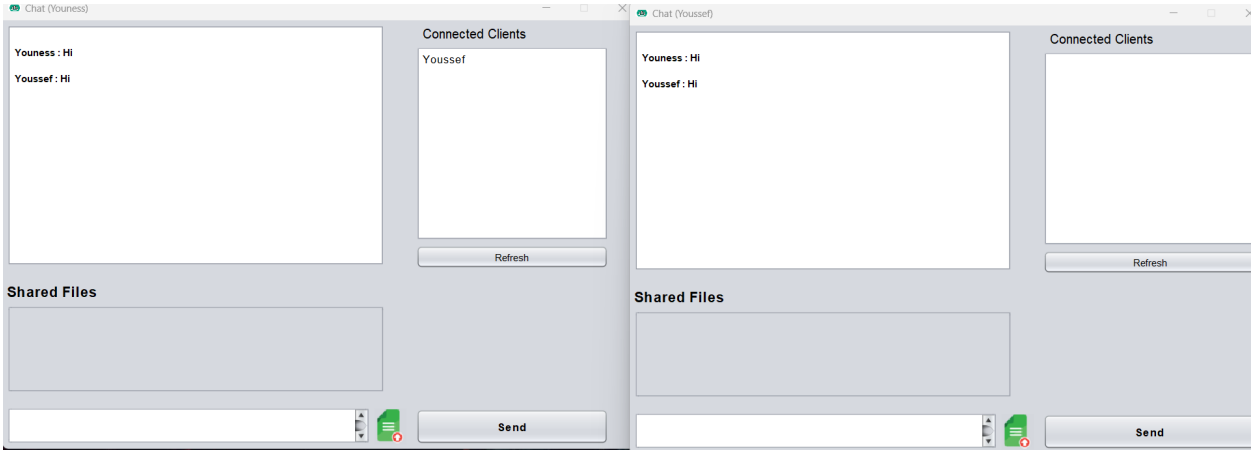
Stack Utilisé

Le projet utilise les technologies et outils suivants :

- **Java RMI** : L'invocation de méthode à distance (RMI) de Java est utilisée pour faciliter la communication entre le serveur de chat et les clients via le réseau. RMI permet aux objets Java d'inviter des méthodes sur des objets distants, permettant une interaction transparente entre les composants distribués.
- **Swing (GUI)** : L'interface utilisateur graphique (GUI) du client de chat est développée en utilisant Swing, une boîte à outils GUI Java. Swing offre un ensemble complet de composants et d'utilitaires pour créer des interfaces utilisateur riches et interactives, permettant la création d'applications visuellement attrayantes et conviviales.







Comment exécuter le code

Pour ChatServer :

1. **Changer de répertoire vers ChatServer :**

```
cd ChatServer
```

- **Compilation :**
  - Changer de répertoire vers le code source :

```
cd src
cd com
```

```
cd remote
cd server
```

- Compiler le fichier Main.java :

```
javac Main.java
```

- **Exécution :**
  - Exécuter la classe Main :

```
java Main
```

Pour ChatClient :

1. **Changer de répertoire vers ChatClient :**

```
cd ChatClient
```

- **Compilation :**
  - Changer de répertoire vers le code source :

```
cd src
cd com
cd remote
cd client
```

- Compiler le fichier LoginView.java :

```
javac LoginView.java
```

- **Exécution :**
  - Exécuter la classe LoginView :

```
java LoginView
```

En suivant ces étapes, vous pourrez compiler et exécuter la classe `Main` dans le répertoire `ChatServer` et la classe `LoginView` dans le répertoire `ChatClient`. Ajustez les chemins de répertoire et les noms de classe si nécessaire en fonction de la structure de votre projet.