

ChatFusion

Rapport de Projet

SDAF Zineb - RAVENDRAN Preamthy

TABLES DE MATIERES

Compilation	3
Lancer le Server	3
Lancer le Client	3
Utilisation	4
Identification	4
Envoie Messages Publics	4
Envoie Messages Privés	4
Envoie Fichiers Privés	4
Fusion	5
Architecture	5
Découpage du Projet	5
Pattern Visitor	5
Difficultés rencontrés	6

Compilation

Lancer le Server

Tout d'abord on se place dans le fichier bin ou dans le fichier out/production/ChatFusion en fonction de votre environnement de développement dans lequel vous aurez compilé (IntelliJ - Eclipse)

Commencé par lancer un ou plusieurs serveur, ServerChatFusion, en lançant la commande,

```
java fr.upem.net.tcp.nonblocking.server.ServerChatFusion  
NameServer Port
```

Ou à l'extérieur du bin

```
java -jar ServerJar.jar NameServer Port
```

Avec **NameServer** correspondant au Nom du Server que nous avons choisi
Et le **Port** correspondant au numéro de port auquel le serveur sera connecté

Lancer le Client

On lance ensuite un ou plusieurs clients, ClientChatFusion, en effectuant la commande suivante:

```
java fr.upem.net.tcp.nonblocking.client.ClientChatFusion Hostname  
Port
```

Ou à l'extérieur du bin

```
Java -jar ClientJar.jar Hostname Port
```

Avec **Hostname** correspondant à l'adresse IP du client
Et le **Port** correspondant au numéro de port du serveur auquel le client souhaite être connecté

Utilisation

Identification

Après s'être connecté au serveur, le client doit indiquer son identifiant, si celui existe déjà dans le serveur auquel il est connecté, alors l'authentification sera refusée et attend que le client entre un identifiant correct.

Si l'identifiant est correct, le client sera donc bien connecté au serveur et un message de confirmation de la connexion s'affiche.

Envoie Messages Publics

Une fois que l'authentification est acceptée au serveur, le client peut envoyer un message public au serveur qui sera diffusé à tous les autres clients authentifiés sur ce même serveur, ainsi qu'à lui-même.

Envoie Messages Privés

Pour envoyer un message privé, le client devra écrire la commande suivante dans la console:

```
@login:server message
```

Où 'login' est le login du client destinataire

Et 'server' est le nom du serveur destinataire

Suivi de 'message' le contenu du message à envoyer

Le message sera ensuite diffusé au client destinataire dans le serveur souhaité.

Envoie Fichiers Privés

Pour envoyer un fichier privé, le client devra écrire la commande suivante dans la console:

```
/login:server file
```

Où 'login' est le login du client destinataire

Et 'server' est le nom du serveur destinataire

Suivi de 'file' le nom du fichier à envoyer

On crée un fichier et écrit dans celui-ci, si le client destinataire reçoit bien la requête attendue.

Fusion

Le serveur possède une console dans laquelle on demande une fusion d'un serveur avec l'adresse `ipaddress:port`, on écrit donc la commande suivante dans la console:

```
FUSION ipaddress port
```

Avec `ipaddress` correspondant à l'adresse `IP`, et `port` le port auquel le serveur est connecté.

Architecture

Découpage du Projet

Le projet contient 5 modules:

- Le Client (incluant le Context Client)
- Les Entités
- Les Readers
- Le Server (incluant le Context Server)
- Les Visitors

Ce découpage permet de pouvoir rajouter / modifier / supprimer des Entités facilement sans impacter le reste du programme.

Pattern Visitor

Nous avons utilisé le Pattern Visitor, permettant d'ajouter des fonctionnalités virtuelles sans avoir à les modifier.

Elle permet aussi de séparer les données exécuté dans la classe `ClientChatFusion` ou dans la classe `ServerChatFusion` utilisant le même objet « Entity ».

Les clients accédant à la structure de l'objet appellent ensuite des méthodes `accept` ou `process` sur l'objet « Entity » concerné.

Ces méthodes délèguent la requête à l'objet visiteur accepté qui peut ensuite exécuter l'opération concernée.

De plus, le code est simplifié et plus clair car chaque visiteurs à une fonctionnalité particulière

Difficultés rencontrés

Lors de ce projet nous avons rencontrés des difficultés tout d'abord sur l'écriture de la RFC, puis dans la mise en place de ce projet.

La compréhension du sujet et de la RFC donné a aussi été difficile pour nous.

La soutenance beta nous a permise de comprendre un peu plus le sujet et de pouvoir implémenter les fonctionnalités proposés.

La partie de la fusion a été la plus grosse difficultés pour fusionner 2 serveur.