TP JMS

Sujet Proposé par D. HAGIMONT

L'objectif de ce TP est de programmer une application répartie en utilisant une implantation de l'interface JMS vue en cours.

Cela consiste à gérer un Forum de discussion pouvant faire intervenir un nombre quelconque d'intervenants utilisant une application Irc.

- Chaque Irc peut émettre des messages qui sont diffusés à l'ensemble des autres Irc
- Les messages ne sont pas mémorisés par le Forum. Seuls les Irc présents sur le Forum reçoivent les messages émis
- Pour émettre, un Irc doit être connecté au Forum

L'application Irc fournie met en place une interface permettant à l'utilisateur de réaliser certaines opérations en activant des boutons :

- Connect : bouton de connexion
- Write : bouton d'envoi d'un message
- Who : bouton pour afficher la liste des utilisateurs connectés
- Leave : bouton pour quitter le forum

1 Installation et test de ActiveMQ

Pour utiliser activeMQ (l'implantation de JMS) :

- récupérez apache-active-MQ : cp /mnt/n7fs/ens/tp_dh/tp-jms/apache-activemq-5.14.0-bin.tar.gz .
- décompresser l'archive apache-activemq-5.14.0-bin.tar.gz
- lancer un daemon activeMQ : bin/activemq start (et stop pour l'arrêter)
- si vous utilisez eclipse, ajouter dans le BuildPath : /activemq-all-5.14.0.jar
- si vous voulez compiler (ou exécuter) en dehors d'eclipse : javac -cp .:/lib/*

Lire le code de l'exemple HelloTopic.java fourni, et testez le.

2 Implémentation de l'application IRC

L'application IRC fournie (à compléter) gère un Forum de discussion pouvant faire intervenir un nombre quelconque d'intervenants.

- Chaque intervenant peut émettre des messages qui sont diffusés à l'ensemble des autres
- Les messages ne sont pas mémorisés par le Forum. Seuls les Irc présents sur le Forum reçoivent les messages émis
- Pour émettre, un Irc doit être connecté au Forum

Le code fourni met en place une interface permettant à l'utilisateur de réaliser certaines opérations en activant des boutons :

- Connect : bouton de connexion
- Write : bouton d'envoi d'un message
- Who : bouton pour afficher la liste des utilisateurs connectés
- Leave : bouton pour quitter le forum

Il implémente aussi une liste d'utilisateurs (Vector users) pour gérer les intervenan connectés.

2.1 Etape 1

On peut commencer par une version minimaliste qui permet d'envoyer des messages et d'afficher les messages reçus. En vous inspirant de HelloTopic.java, coder les opérations de connexion, emission et réception de messages :

- Complétez les classes connectListener, writeListener, et leaveListener
- Codez la méthode MessageListener, qui sera activée lors de le réception d'un message

Aucune autre modification n'est à faire.

On utilisera des StreamMessage qui permettent des échanges de données en flux de valeurs de type primaire. Ils permettent dans notre cas d'envoyer et de recevoir des messages composés de différents champs : une entête (qui indique le type du message), suivie du corps du message.

```
        StreamMessage smsg = session.createStreamMessage();
        smsg.writeString(entete);
        smsg.writeString(corps_du_message);
        producer.send(smsg);
        ...
```

```
public void onMessage(Message msg)  {
        try {
                StreamMessage smsg = (StreamMessage)msg;
                String entete = smsg.readString();
                // smsg.readString() pour récupérer le champs suivant
                ...
```

2.2 Etape 2

Pour gérer toutes les fonctionnalités de l'application, on aura besoin d'identifier différents types de messages :

- message de connexion
- message à afficher
- message de déconnexion
- message permettant à tout utilistaeur présent d'informer de sa présence

Elaborez une stratégie permettant de gérer la liste des utilisateurs connectés, en répondant à la question suivante : que faire lorsqu'on reçoit :

- un message de connexion
- un message à afficher
- un message de déconnexion
- un message indiquant la présence d'un utilisateur

Complétez le code pour implanter une solution gérant la liste des intervenants connectés

2.3 Etape 3

Si on souhaite recevoir les messages émis lorsqu'on a quitté l'application Irc et que l'on s'y reconnecte plus tard, il faut initier une connexion durable.

```
//Connexion non durable
consumer = session.createConsumer(topic);
```

```
//Connexion durable
connection.setClientID(myName);
consumer = session.createDurableSubscriber(topic, myName);
```

Modifiez le code pour implanter des connexions durables.