

JMS exercise

Daniel Hagimont

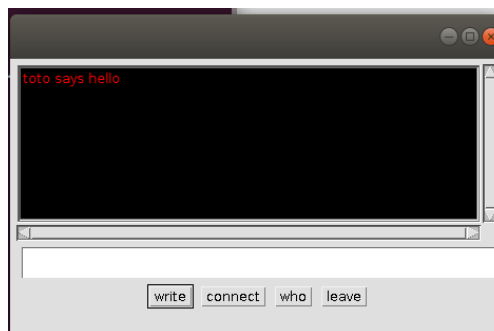
Daniel.Hagimont@enseeiht.fr

USTH

March 2020

The objective of this lab is to practice with JMS (Java Message Service) and to implement a simple chat application.

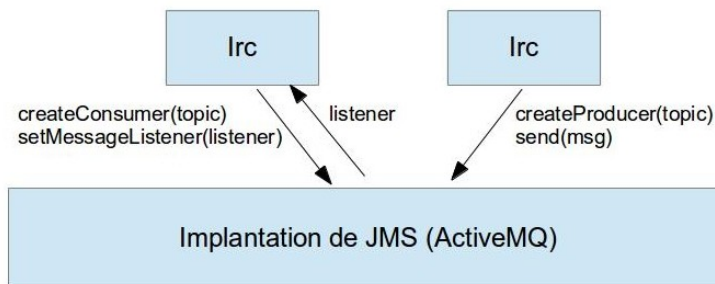
This application (called Irc) manages a Forum of chat, allowing many users to talk together. All instances of Irc, located on different nodes, can connect to the Forum. Then, they can emit messages which are broadcast to all the instances connected with the Forum.



Above is the GUI of the Irc application. It has 4 buttons for which you have callbacks in the code, that you can adapt to implement communication between Irc instances. These buttons have the following meanings:

- Connect : allows the Irc instance to connect to the Forum.
- Leave : allows the Irc instance to disconnect
- Write : allows the Irc instance to emit a message (which was typed in the blank bar)
- Who : allows the Irc instance to show the list of participants in the Forum.

The Irc application is only composed of the Irc.java file. This class includes the GUI of the application, where you should adapt the button callbacks to implement communication using JMS. We use JMS in Publish/Subscribe mode to implement Irc.



In order to help starting, a little example ([HelloTopic.java](#)) of JMS code is given. You can compile and test it.

The implementation of JMS that we use is activeMQ. To use it :

- Uncompress the archive (apache-activemq-5.14.0-bin.tar.gz) in a directory <activemq_path>
- To start or stop activeMQ (daemon) :
 - <activemq_path>/bin/activemq start
 - <activemq_path>/bin/activemq stop
- For eclipse, add in your BuildPath : <activemq_path>/activemq-all-5.14.0.jar
- To compile or execute from the command line :
 - javac -cp .:<activemq_path>/activemq-all-5.14.0.jar <fichier source java>
 - java -cp .:<activemq_path>/activemq-all-5.14.0.jar < java class>

- 1) Look at the source, compile and execute the HelloTopic.java example
- 2) Implement the Connect (subscribe to a topic) and Write (on the topic) operations, and the code to receive messages and print them in the window.
- 3) Propose a solution for the Who operation
- 4) Add the management of "durable" connections. With durable connection, if you leave Irc and reconnect later, you don't lose message, i.e. you receive the messages that were sent while you were disconnected. Look on the internet for documentation.