

# Chat rooms 2016-2017

L'objectif de ce TP est de développer un centre de chambre de discussion.

## 1.Importation

Pour utiliser le zip, il faut dans l'ordre :  
 Télécharger le zip et le laisser compacté  
 Ouvrir Eclipse  
 Demander à eclipse d'importer un projet  
 Dans la fenêtre choisir un projet existant  
 Puis vous aurez deux options, faut prendre celle qui suggère d'importer depuis un fichier zip ☺  
 Si vous avez bien fait, vous aurez trois projets dans votre espace de travail.

## 2.Structure

Vous avez (maintenant et par les siècles à venir) trois projets dans votre espace de travail.

- fr.ensisa.hassenforder.chatrooms.client.empty
- fr.ensisa.hassenforder.chatrooms.common.empty
- fr.ensisa.hassenforder.chatrooms.server.empty

Client est l'interface graphique qui contient deux classes avec une méthode main mais c'est uniquement la classe appelée Application qu'il faut utiliser.  
 Server est le server sans interface, il y a également une classe Application avec une méthode main.  
 Common est ce qui est commun (bizarre, comme c'est bizarre, ... vous avez dit bizarre, comme c'est étrange). Les deux projets client et server référencent Common.

## 3.Nouvelles contraintes liées à l'IHM

Une IHM ne supporte pas de bloquer l'exécution de la tâche principale pour une opération longue. Faire une requête réseau est une opération longue. Donc, j'ai mis des 'protections'.  
 Quand on clique sur un bouton on va lancer la requête dans une tache dédiée et quand l'opération est terminée en met à jour l'interface graphique dans la tache dédiée. Pour simplifier, chaque interaction appelle un listener (interface) enregistré, par défaut c'est la classe Application.  
 Donc pour vous c'est dans application que se trouve le code important.

## 4.Nouvelles contraintes liées au réseau

Selon votre vision du protocole vous avez ou non détecté que votre socket devra être écoutable/écrivable à volonté depuis chaque extrémité. Ce n'est pas nécessairement un problème mais cela pourrait complexifier nettement le code car il faudra toujours associer les réponses aux questions et ne jamais se mélanger les pinceaux. En TP... une gageure.  
 Je vous ai donc proposé un code avec 2 sockets actives indépendamment. Une socket de commandes et une socket de messages. Commandes est pilotée par l'IHM pour envoyer ces requêtes et avoir (éventuellement) ses réponses. Messages est pilotée par le serveur pour envoyer les messages vers le client.  
 Le corollaire de cette décision... Il n'y a pas une session, un reader et un writer par application (donc une paire de chaque) mais deux paires par application. Ce qui fait que vous devrez intellectuellement bien comprendre quel triplet vous mettez à jour (session/reader/writer). Faites attention, une modification d'un triplet avec un esprit embrumé (1<sup>er</sup> Janvier) vous précipite vers une catastrophe.