


PROGRAMMER UNE APPLICATION CLIENT-SERVEUR TCP

On utilisera le langage JAVA pour fabriquer une application client-serveur.

On désire réaliser une application constituée d'un client en JAVA FXML et d'un serveur en JAVA.

1. Côté serveur :

Côté serveur le logiciel est capable de répondre à un jeu de requêtes fermées provenant du réseau. Ce jeu de requêtes, leur forme, la forme des réponses attendues, ainsi que l'ordre des échanges (il y a-t-il un message de bienvenue et une invite de commande) sont des choses prédéfinies comme une convention entre le client et le serveur et sont figées par la suite. Cela constitue le protocole de "mon service".

1-1 Travail à faire :

Le serveur enverra un message de bienvenue au début ainsi qu'une invite de commande à la fin de chaque réponse.

Les requêtes valides et les réponses associées sont les suivantes :

- Si cette requête est « **HELLO »**, le serveur **répond un message de bienvenue**.
- Si cette requête est « **TIME »**, le serveur **répond la date qu'il est (ou l'heure, ou les deux au choix)**.
- Si cette requête est « **ECHO …une phrase…** », le serveur **répond en répétant la phrase**.
- Si cette requête est « **YOU** » ou « **WHOAREYOU?** », le serveur **renvoi sa socket** (IP@ et port).
- Si cette requête est « **ME** » ou « **WHOAMI**? », le serveur **renvoi la socket du client** (IP@ et port) qu'i récupère dans la connexion établie par le client.
- Si cette requête est « **FIN** », le client est déconnecté.

Selon ce que contient la requête, le serveur fournit une réponse appropriée : l'algorithme est du type

```
if  (Requete == "HELLO") {
      Reponse = "Vous êtes connectés au serveur SUPERSERVEUR ....");
 }

if  (Requete == "TIME") {
      Reponse = "Voici la date et l'heure" +  MaFonctionRecupDateHeure(…);
}

etc...
```

Pour vous aider vous avez à disposition pour tester votre serveur:

- Un client fonctionnel « ***Client_TCP_Prof.exe***» .

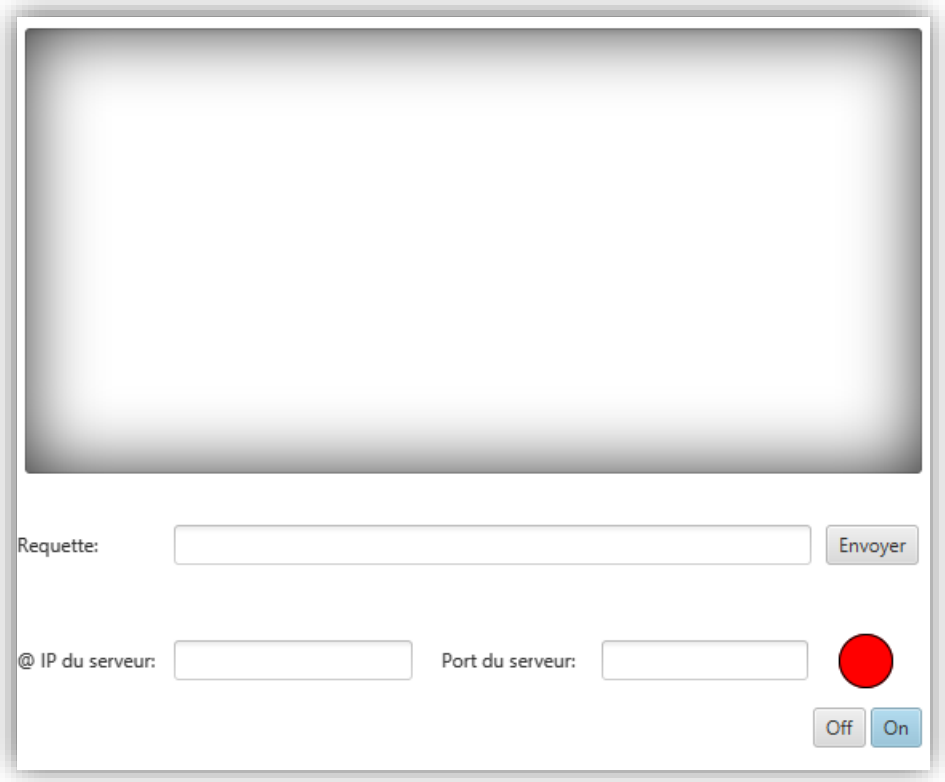
2. Côté client graphique :

Côté client le logiciel est capable de demander où se connecter (@IP et port), puis d'afficher le message d'accueil reçu du serveur après connexion.

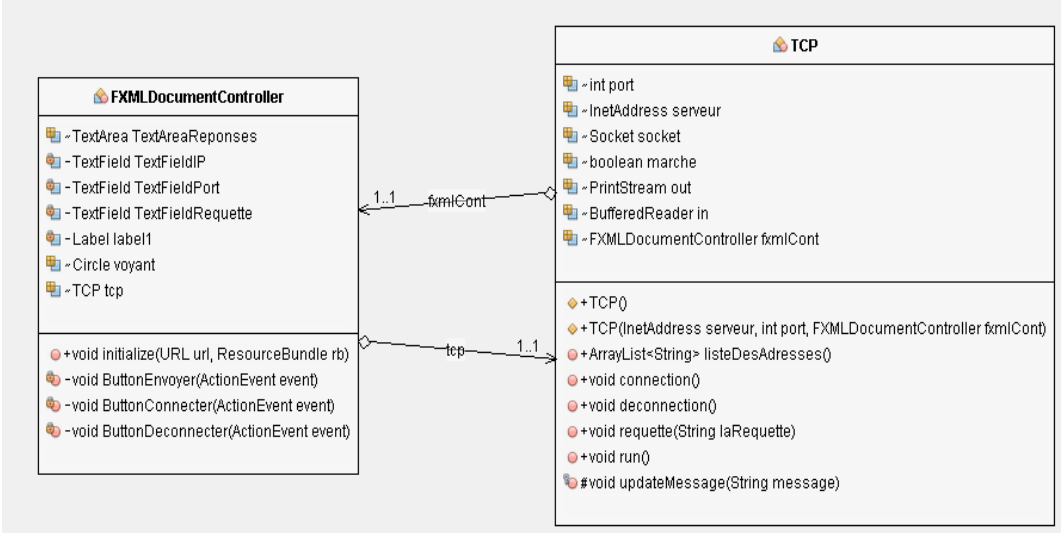
Alors il rentre dans une boucle de saisie d'une requête et d'affichage de la réponse reçue du serveur.

La sortie se fait avec la requête « ***exit*** » qui est aussi interceptée en local ou le ***Button OFF***.

2-1 L'interface graphique proposé est la suivante :



2-2 Organisation du projet :



```
/*
Pour déclencher une opération graphique en dehors du thread graphique  utiliser
javafx.application.Platform.runLater(java.lang.Runnable)
Cette méthode permet d'éxécuter le code du runnable par le thread graphique de JavaFX.
*/

protected void updateMessage(String message) {
    Platform.runLater(() -> fxmlCont.TextAreaReponses.appendText("    MESSAGE SERVEUR >  \n      " + message + "\n"));
}
```

PROGRAMMER UNE APPLICATION CLIENT-SERVEUR UDP

Reprenez le client serveur avec des datagramme UDP pour la couche transport.
Pour votre client/serveur vous pouvez garder le client/serveur TCP en y rajoutant un commutateur TCP◄►UDP (en console et en FXML).