

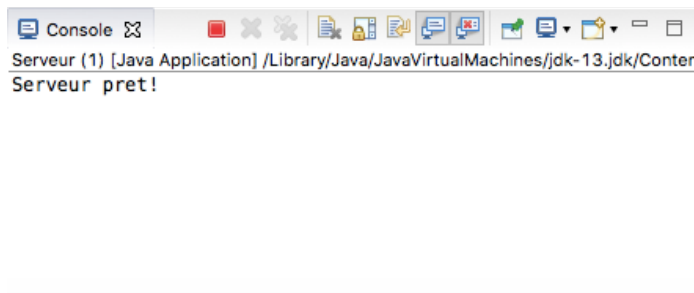
### Explication RMI :

Tout d'abord, j'ai repris votre structure, à savoir deux projets différents avec un qui gère le côté client et l'autre le côté serveur.

Ces deux projets sont reliés grâce à l'interface InterfaceInft.

J'ai d'abord commencé à comprendre le fonctionnement de la RMI qui nous permet ici d'appeler des méthodes dans des classes et des projets différents.

Comme on peut le voir sur la capture d'écran ci-dessous, le serveur était opérationnel.



Pour créer un chat, je me suis dit que la méthode la plus simple serait de créer deux listes, une au niveau du serveur qui gère les messages envoyés : *Message* et l'autre au niveau du client qui est modifiée à chaque nouvelle entrée : *ListeEntree*.

```
private ArrayList<String> ListeEntree = new ArrayList<String>() ;

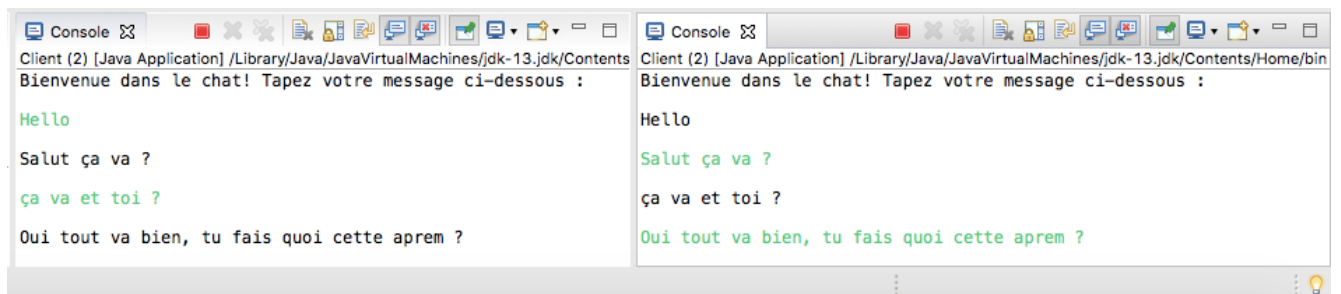
public ArrayList<String> getListeEntree () {
    return ListeEntree ;
}

public void setListeEntree (ArrayList<String> ListeEntree) {
    this.ListeEntree = ListeEntree ;
}
```

J'ai ensuite mis en place deux thread. L'un, nommé Polling, qui récupère les messages pour les traiter ensuite et l'autre, Sender, qui gère le renvoi du message à l'autre client.

La comparaison des deux listes se fait en fonction de la taille de ces listes. Ainsi dès qu'un nouveau message est entré dans la liste côté client, le serveur se rend compte que sa liste est plus courte et procède à la mise à jour de sa liste.

On peut apercevoir ci-dessous que le chat fonctionne correctement après plusieurs modifications visant à ne pas réafficher chez le client le message qu'il venait d'afficher et à ne pas afficher le nouveau message à l'infinie :



Cependant, j'ai perdu beaucoup de temps à trouver une erreur « d'espaces » en trop entre mes deux interfaces qui n'étaient donc pas similaires à 100% ce qui empêchait la liaison entre mes différentes méthodes à travers mes classes.

Cette perte de temps un peu ridicule m'a empêché de pouvoir finir mes travaux sur le chat version callbacks.