

Java RMI

Téléchargez le fichier RMI.tar.gz dès la page web du cours:

[RMI.tar.gz](#)

Exercice 1.

a) Extraire le contenu du fichier dans un dossier de votre choix. Examinez les fichiers, qui correspondent à l'exemple vu en cours. Identifiez quels fichiers correspondent aux composants suivants:

- Définition du service
- Implémentation du service
- Implémentation du client
- Implémentation du serveur

b) Examinez le code client et serveur: sur quelle adresse et porte la connexion rmi aura lieu? S'agit-il d'une interaction locale ou à distance, selon vous?

c) Compilez les fichiers .java

d) [facultatif] Créez le *stub* pour les applications (rappel: **rmic** implémentation du service)

e) Modifiez le fichier de police client1.policy de façon que le codebase corresponde au dossier où vos fichiers sont stockés: par exemple, si vos fichiers sont stockés dans le dossier "/users/12344321/Reseaux/TP3/RMI", le fichier de police sera le suivant:

```
grant codeBase "file:/users/12344321/Reseaux/TP3/RMI/" {  
    permission java.security.AllPermission;  
};
```

f) Maintenant vous êtes prêts pour lancer le registre rmi: d'abord, vérifiez qu'il n'y a pas de processus qui utilise la porte 1099. Vous pouvez utiliser la commande **netstat** (par exemple, `netstat -nap | grep 1099`).

Lancez `rmiregistry` avec l'option `-J-Djava.security.policy=client1.policy` .

Vérifiez que le registre est bien visible à la porte 1099 (`netstat ...`).

g) Lancez le serveur avec `java <nom_programme_serveur>`

`-Djava.security.policy=client1.policy` (s'il y a des problèmes de permissions, vérifiez le fichier `client1.policy`).

h) Vérifiez le fonctionnement du programme avec `java <nom_programme_client> <chaine_de_test>`. Testez le fonctionnement client-serveur en utilisant différents clients (ouvrez des nouveaux terminaux).

i) Essayez de modifier le code du client pour contacter le serveur sur la machine du votre voisin. Le nom d'une machine du labo est de la forme `rXXXXY` ou `sXXXXY` (vous le trouvez dans la console sur la ligne de commande, après votre nom d'utilisateur et le symbole `@`), vous pouvez utiliser celui-là ou l'adresse ip correspondant (vous pouvez l'afficher avec `ping <nom_de_machine>`).

Exercice 2 (facultatif - pour les plus avancés).

Modifiez votre projet de banque pour fonctionner avec RMI:

- La banque doit être un service et ses méthodes doivent être des méthodes RMI (donc qui lèvent une exception `java.rmi.RemoteException`)
- Le client fonctionne comme un guichet automatique bancaire

Vérifiez le bon fonctionnement des transactions, en particulier si plusieurs utilisateurs ont accès au même compte dans le même temps.

Plusieurs solutions sont possibles en cas de conflit::

- Verrou sur le compte
- Rejeter toute connexion du même utilisateur