# Projet de Données Réparties : Un service de partage d'objets répartis et dupliqués et robustes en Java Etape 2

Justin BARBIER
Tom BERTRAND
Maël MATHURIN

Département Sciences du Numérique
Deuxième année - Parcours Systèmes Logiciels
2022 - 2023

.

# 1 Ajouts par rapport à l'étape précédente

— Ajout de méthodes track() et leave_track() aux différentes classes afin de permettre à un client de "track" un objet et effectuer une lecture de l'objet à chaque changement qu'il subit, ajout également de la méthode trace() au Client et SharedObject qui permet de mettre à jour un objet partagé "tracké".

— Ajout d'une liste de Client "tracers" dans la classe ServerObject afin d'appeler la méthode trace() sur le Client qui le propage au SharedObject.

— La fonction callBack a appeller est maintenant donné par l'utilisateur en utilisant la méthode setCallBack(CallBack callback), et peut l'enlever en utilisant removeCallBack.

— Le callBack est maintenant indépendant du subscribe ou du track : on peut ne pas avoir de callBack en étant subscribe ou en track. Etre en subscribe sans avoir de callBack ne sert cependant à rien. Ne pas avoir de callBack en étant en track permet de récupérer le nouvel état de l'objet et de passer notre état en RLC, donc avoir une lecture plus rapide de l'objet.

— Dans la fonction unlock, lorsque l'utilisateur a fini d'ecrire, le nouvel état de l'objet est envoyé au serveur, afin que celui-ci en garde une copie et l'envoie à tout les clients qui "track" l'objet (hors celui qui vient de l'écrire) par l'intermédiaire de la méthode trace(Objet obj) présent dans la classe SharedObject.

# 2 Explication d'architecture

— Comme pour le subscribe, le Client a le choix de track ou non à n'importe quel moment. Un client ne peut pas être â la fois subscribe et tracker l'objet : un appel à subscribe supprime le track (si il track l'objet) et inversement.

— Lorsqu'on est en track et que l'objet est actualisé, en plus de mettre à jour l'état local de l'objet, la fonction callBack est appellé.

— Avoir une liste de Client trackers stockée dans le ServerObject nous permet d'avoir un lien direct avec les Client concernés afin de faire appel au Callback qui lui est connu par le SharedObject correspondant. Le ServerObject appel à distance le SharedObject afin qu'il fasse son callBack.

— Pour changer le callBack en fonction de si on est subscribe ou en track, il faut la redéfinir à chaque fois qu'on change de mode par la méthode setCallBack(CallBack callback). Il y a donc 2 Callbacks différents dans IRC, un qui actualise le compteur pour le sbuscribe, et un qui affiche le text pour le track.

— Dans IRC : ajout des boutons pour appeler track et leave_track.

# 3 Démonstration

Dans cette section, nous présenterons des captures d'écran de notre démonstration accompagnées d'explications

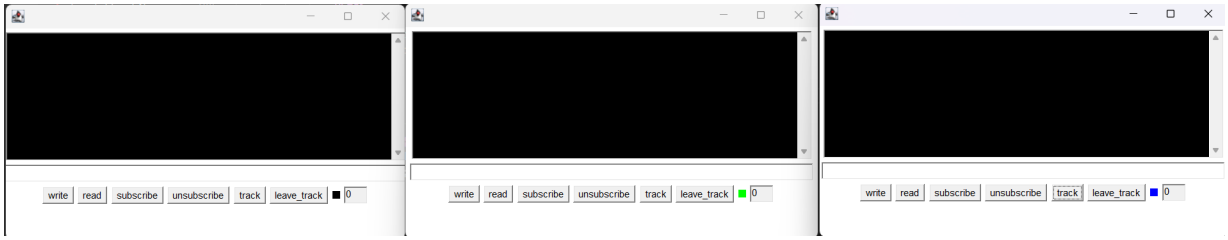


FIGURE 1 – Situation initiale des 3 clients

Dans cette démonstration, nous utiliserons 3 clients lancés simultanément. Un "writer", un "subscriber" et un "tracker".

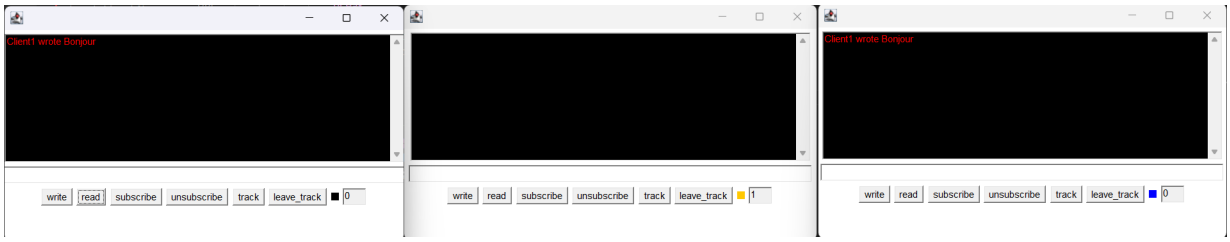


FIGURE 2 – 1 clients abonné, 1 tracker

Premièrement, le client 2 s'abonne à l'objet partagé et le 3 le track. L'indicateur de 2 devient orange car il n'est pas à jour et le 3 effectue automatiquement une lecture.

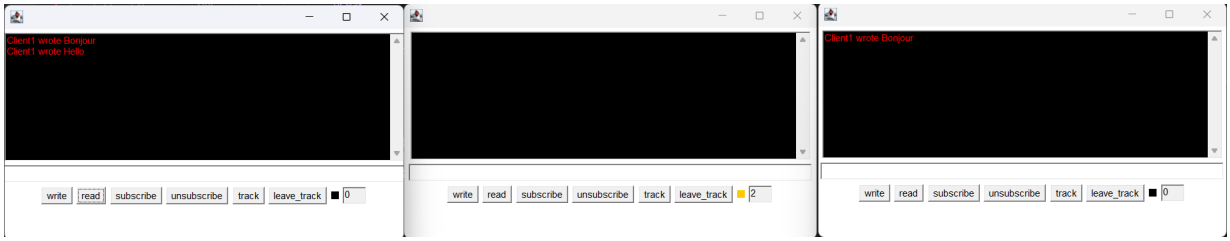


FIGURE 3 – Client 3 leave track

Client 1 réécrit et 3 ne lit pas car il a leave track

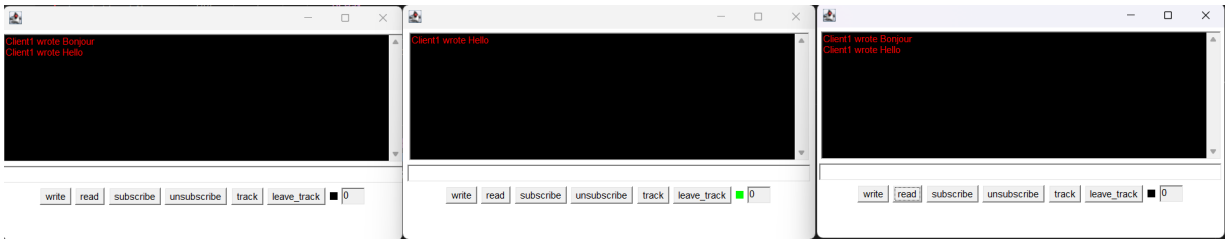


FIGURE 4 – Client 2 lit l'objet

Client 2 effectue une lecture, ce qui lui permet d'afficher la dernière valeur de l'objet et change l'indicateur au vert, le compteur passe à 0. Client 3 peut toujours accéder à la dernière valeur de l'objet en faisant une lecture manuelle.

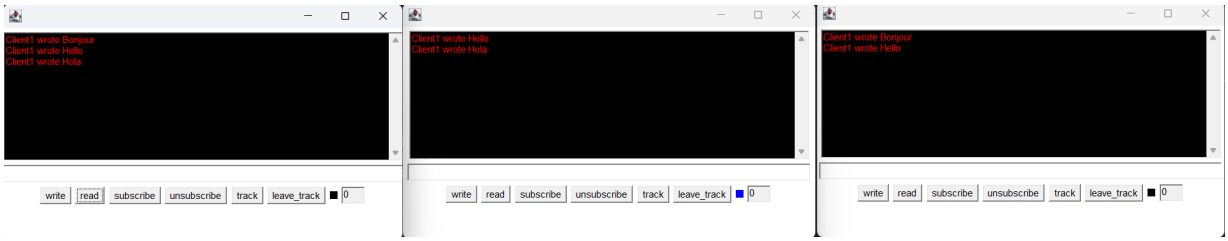


FIGURE 5 – 3 écritures non lues

Client 2 passe de l'état subscribed à tracker en cliquant sur track sans cliquer sur unsubscribe : on passe d'un état à l'autre directement.

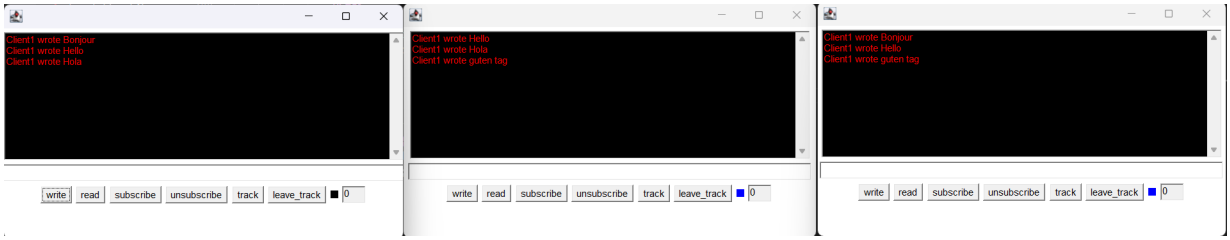


FIGURE 6 – Client 3 effectue une lecture

Client 3 peut re-track et la lecture est automatique.

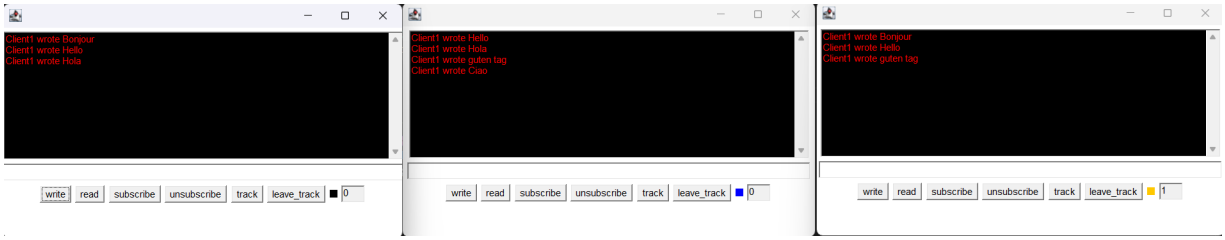


FIGURE 7 – Lecture par Client 2

On passe également de tracker à subscriber en une opération.

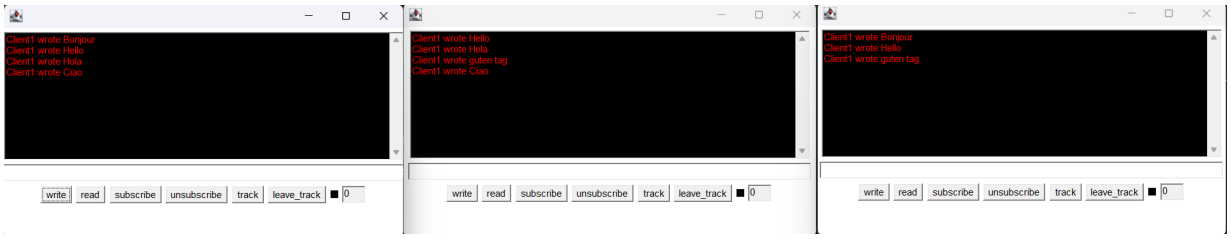


FIGURE 8 – Client 3 écrit

Les Clients 2 et 3 peuvent quitter leur état subscriber ou tracker.

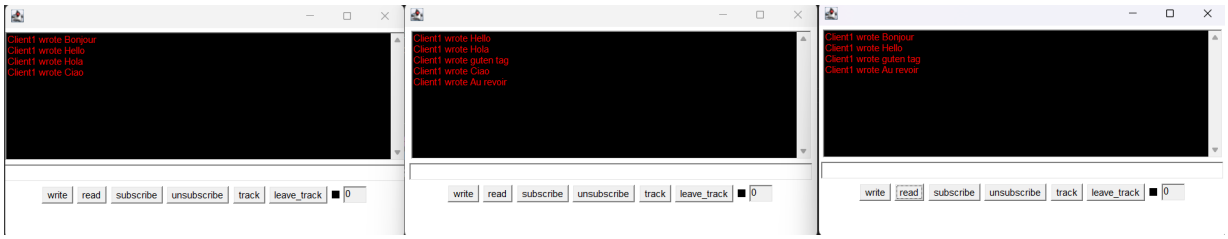


FIGURE 9 – Lectures par les clients

L'objet a bien enregistré les modifications, ce qui termine notre démonstration.