

Rapport PDR S8 Etape 1

Jed Moutahir, Steven Zheng

February 2023

1 Modifications apportées

L'objet serveur (*ServerObject*) contient une nouvelle liste. Elle recense les abonnés à cet objet.

La logique implémentée est la suivante :

Si un client s'abonne à un *SharedObject*. Le *SharedObject* enregistre le client ainsi que son *callback*. La requête est ensuite transmise au serveur qui cherche le *ServerObject* associé à l'identifiant de l'objet et demande un abonnement au *ServerObject*.

Lors d'une modification, si *unlock* est appelé, le client indique au serveur qu'il a effectué une modification : *notifier* est donc appelé. Le *ServerObject* se charge de transmettre le message en appelant le *callback* sur tous les abonnés à cet objet.

L'objet serveur fournit de nouvelles méthodes :

- `public void abonner(Client_itf client)` enregistre le client comme abonné.
- `public void desabonner(Client_itf client)` retire le client des abonnés.
- `public void notifier()` notifie tous les abonnés de la modification de cet objet.

Le client fournit de nouvelles méthodes :

- `public static void abonner(int id)` permet de s'abonner à un objet.
- `public static void desabonner(int id)` permet de se desabonner à un objet.
- `public void callBack(int id)` qui permet de rappeler le client abonné.
- `public static void notifier(int id)` qui permet de notifier après avoir modifier l'objet.

Le serveur fournit de nouvelles méthodes :

- `public void abonner(Client_itf client, int id)` transmet la requête d'abonnement au bon *ServerObject*.
- `public void desabonner(Client_itf client, int id)` transmet la requête de désabonnement au bon *ServerObject*.
- `public void notifier(int id)` transmet la requête de notification au bon *ServerObject*.

L'objet partagé fournit de nouvelles méthodes :

- `public void abonner(Callback_itf callback)` permet de s'abonner à cet objet.
- `public void desabonner(Callback_itf callback)` permet de se désabonner à cet objet.
- `public void callback()` exécute le callback sur cet objet.

Une nouvelle interface *Callback_itf* a été ajoutée. Un exemple d'implémentation est visible sur *Irc*. (ligne 36 de *Irc.java*).

2 Démonstration

Une vidéo nommée *demo.mp4* montre le fonctionnement du nouvel Irc pour deux clients. On peut voir qu'après avoir appuyé sur *write*, le callback des deux clients est appelé (Le message est visible dans la console). Une autre implémentation possible aurait été d'effectuer la même tâche que lorsque le bouton *read* est appuyé.