

Architecture et implémentations JarRetCLIENT

L'implémentation de la boucle principale

- 1) Formation et envoi de la requête GET demandant au serveur une task à accomplir
- 2) Réception de la réponse du serveur. Création d'un objet HttpHeaders, lecture de contenu JSON
- 3) Création d'une map dans ServerAnswer en parsant le JSON renvoyé par le Serveur. Toutes les informations envoyées par le Serveur sont stockées dans une instance de **ServerAnswer**.
- 4) Le Client vérifie si le serveur lui a envoyé un **Come Back In Seconds**.
- 5) Le Client crée l'instance demandée par le Serveur avec la méthode **JarRetClient::returnAWorker** (voir plus bas)
- 6) Génération du CONTENU de la requête POST (voir plus bas)
- 7) Envoi de la requête POST au serveur
- 8) Écoute de la réponse du serveur
- 9) Retour à l'étape 1.

Instanciation d'un Worker

```
HashMap<Long, HashMap<String, Worker>> workers = new HashMap<>();  
private Worker returnAWorker(Long jobId, String workerVersion, ServerAnswer  
serverAnswer) ;
```

Cette méthode va chercher dans la HashMap **workers**, une instance du Worker correspondant au **jobId** et à la version du Worker demandé : **workerVersion**. Si on trouve la bonne instance, c'est cette instance qui sera utilisée pour effectuer le calcul avec **Worker::compute**. Sinon on instancie un nouveau Worker à l'aide de la méthode **WorkerFactory::getWorker**.

Génération du CONTENU JSON de la requête POST

On peut diviser le CONTENU JSON en deux parties:

- La première partie qui contient les informations envoyées par le serveur et stockées dans la classe **ServerAnswer**. La méthode **JarRetClient::createContentWithoutLastLine** génère cette partie en utilisant les getter de **ServerAnswer**.

- La deuxième partie qui correspond à la dernière ligne du CONTENU JSON qui peut indiquer le résultat du calcul « Answer: ... » ou alors un message d'erreur destiné au serveur « Error: ... ».

C'est la méthode **private static String computeLastLine**(Worker worker, int taskNumber) qui gère le calcul de cette dernière ligne. Elle tente d'effectuer le calcul demandé avec **Worker::compute**. La méthode analyse la réponse de compute, si une erreur est détectée (Json invalide, Json imbriqué ou Json trop long) la dernière ligne sera « Error : ... » , sinon elle sera « Answer : { ... réponse de compute... } »