# Serveur JarRet
# Documentation

## Description du serveur JarRet:

Implémenté en utilisant le protocole HTTP via TCP en mode non-bloquant.

Il utilise un fichier JSON situé dans *resources/* pour lister tous les jobs qu'on lui fait faire.

Il en distribuera des tâches, suivant la priorité de chaque job, aux clients qui se connectent, plus la priorité du job est basse, moins on en distribuera de tâches (par rapport à l'ensemble des jobs).

Lors d'une communication avec un client, le serveur peut être dans quatre états différents :
- *CONNECTION* : Le client envoie une requête au serveur pour recevoir une tâche
- *TASK* : Le serveur répond en envoyant une tâche.
- *RESPONSE* : Le client traite la tâche et en envoie la réponse au serveur, qui la traite.
- *END* : Fin de communication entre le serveur et le client pour la tâche donnée, le serveur envoie au client si la tâche a bien été reçue ou non.

Lorsque tous les jobs sont finis, le serveur renvoie, au lieu de *TASK*, un paquet disant de demander une tâche plus tard, dans 300 secondes.

On utilisera une classe *Context*, qui sera attachée aux *SelectionKey* du *Selector* du serveur, pour gérer la communication entre le serveur et un client.

Il est également possible de taper des commandes (*STOP*, *SHUTDOWN*, *SHOW*), dont l'entrée se fait via un *Scanner* d'une autre *Thread*. Une fois une commande reçue, il la place dans une *BlockingQueue*, et réveille la *Thread* principale bloquée sur un *Selector::select* pour traiter la commande dès que possible s'il était en train d'attendre.

## Arborescence du serveur :

,
- Package : upem.jarret.server :

JarRetServer : le serveur en lui-même, qui se lance à l'aide de la méthode *launch()*.

```
|____ JobMonitor :   un objet monitorant un Job, pouvant savoir quelle tâche a été
|     |              effectuée, donner des tâches, et écrire les résultats dans un
|     |              fichier.
|     |____ Job :    un objet contenant l'ensemble des informations parsées dans le fichier
|                    JSON.
|____ Context :      Gère l'ensemble des informations liées à la communication avec
                     un client.
```

- Package : upem.jarret.http :

Utilisé par le serveur, ainsi que le client, pour gérer les communications HTTP.

```
<<HTTPReader>> :   Lit des octets reçus via le protocole HTTP.
|____ <<HTTP Header>> :   Lit les informations du header et en stocke la réponse et les
                          différents champs.
```