

Installation & utilisation de BDD NoSQL

Redis (Remote Dictionary Server, clé-valeur)

Redis (Remote Dictionary Server, clé-valeur)

- De base, fonctionne sous Linux; devrait donc être installé sur une machine Linux, ou un sous-système Linux :
 - activez la fonctionnalité Windows suivante : « *Sous-système Windows pour Linux* ».
 - allez sur le Microsoft Store pour installer *Ubuntu 24.04 LTS*
 - lancez *Ubuntu*, configurez-vous un compte, puis exécutez les instructions suivantes pour tout mettre-à-jour :
 - `sudo apt-add-repository ppa:redislabs/redis`
 - `sudo apt-get update` / `sudo apt-get upgrade`
 - `sudo apt-get install redis-server`
 - puis lancez le server : `redis-server` ou démarrez un service : `sudo service redis-server start`
 - dans une nouvelle fenêtre Ubuntu, lancez un client redis : `redis-cli`
 - où vous pouvez tester les commandes Redis : `set key value` / `set nom Stettler` / `get nom`
- Sinon, vous pouvez installer une (très) ancienne version de Redis pour Windows : *Redis-x64-3.2.100*
 - puis lancez le server : `redis-server.exe` ainsi qu'un client : `redis-cli.exe`
 - où vous pouvez tester les commandes Redis : `set key value` / `set nom Stettler` / `get nom`
- Si vous n'utilisez pas Maven pour gérer les dépendances de votre projet (*bibliothèques externes*), téléchargez un driver redis pour java : *jedis.jar* contenant des dépendances (*slf4j*, ...) incluses dans le zip :
 - vous pouvez le trouver sur <https://jar-download.com/artifacts/redis.clients/jedis>
- Dans un nouveau projet IntelliJ, créez un sous-répertoire *lib* => copiez les drivers (*fichiers.jar*) dans *lib*
- Et indiquez qu'il s'agit d'une librairie Java à utiliser dans votre projet :
`File => ProjectStructure => Libraries => + Java => lib`
- Dans votre application Java, vous pouvez avoir une instance *jedis* : `Jedis jedis = new Jedis();` vous permettant de lancer les commandes : `jedis.set("nom","Stettler");` ou `jedis.get("nom");`

TP-9 - Redis

- Installez une version de Redis sur votre machine.
- Téléchargez un driver redis pour java.
- Développez une petite application Java effectuant des lectures/écritures de données dans une bdd Redis.
- Étudiez les commandes *mset,mget,del*, les listes (*lpush,lrange,linsert,lrem*), sets(*sadd,smembers,sismember,srem*), hash(*hset,hget,hdel,hgetall,hvals,hkeys*), ainsi que *expire,ttl*
- Étudiez également différentes façons de stocker des *Objects* dans Redis : par ex. *Gson.toJson / fromJson*.
- **L'objectif étant de connaître les possibilités de Redis, voir les différentes façons de 'structurer' les données stockées, savoir convertir des classes Java afin de stocker des instances d'objet dans Redis, et être capable de comprendre et utiliser les méthodes principales de Redis (de la classe *Jedis*).**
- Vous pouvez par exemple reprendre un ancien TP (*Compétition*), et voir comment stocker (et retrouver) les participants aux compétitions dans Redis !