

# TP Clé-Valeur sous Redis et Cassandra

## TP2 — Manipulation d'une Base NoSQL Clé / Valeur avec Redis

Dans ce TP, vous allez manipuler une base de données Redis, représentative du modèle clé / valeur.

Redis est très utilisé comme cache, stockage temporaire, ou base temps réel dans les architectures modernes.

Ce TP a pour objectif de vous faire comprendre :

- Le fonctionnement du modèle clé / valeur
- Les principales commandes Redis
- La gestion des clés, des valeurs et de leur durée de vie
- Les structures de données Redis (Hash, List)
- Les transactions Redis

### Exercice — Création et renommage de clés

1. Créer une clé User avec une valeur de votre choix, puis la renommer en User:1.
2. Créer plusieurs clés utilisateurs (User:2, User:3, User:4) avec des valeurs différentes. Lister toutes les clés existantes, puis uniquement celles commençant par User.
3. Afficher les valeurs de plusieurs clés en une seule opération.
4. Modifier la valeur d'une clé existante en ajoutant du texte à la fin de sa valeur.
5. Tester une commande permettant de créer une clé uniquement si elle n'existe pas déjà, et expliquer le résultat obtenu.
6. Créer une clé temporaire User:Temp avec une durée de vie limitée. Vérifier son temps de vie restant, puis observer ce qu'il se passe après expiration.
7. Simuler un utilisateur en créant plusieurs clés liées (User:6, User:6:City, User:6:Age, User:6:Activity) dont au moins une possède une durée de vie limitée.
8. Tester une opération permettant de créer plusieurs clés de manière atomique uniquement si aucune d'elles n'existe déjà, et analyser le comportement obtenu.

9. Créer un utilisateur en utilisant une structure de type Hash, puis afficher la liste des champs et des valeurs.
10. Créer une liste Redis, y insérer plusieurs valeurs, puis effectuer ces opérations dans une transaction Redis.

## TP3 — Manipulation d'une Base Cassandra

### Objectif

Mettre en œuvre une base de données Cassandra afin de comprendre le fonctionnement d'une base NoSQL distribuée orientée colonnes. L'objectif de ce TP est de manipuler le langage CQL, de créer un schéma de données simple, d'insérer et de consulter des données, et de comprendre le rôle de la clé de partition dans la distribution des données.

### Architecture attendue

La base Cassandra sera déployée via Docker sur la machine locale. Les manipulations se feront à l'aide de l'outil **cqlsh**. Un keyspace nommé **mobility** sera créé, contenant une table destinée à stocker l'historique de disponibilité des stations Vélib.

### Fonctionnement

Le travail consiste à démarrer Cassandra, à créer le keyspace et la table **velib\_status**, puis à insérer plusieurs enregistrements correspondant à des stations et à différentes dates. Les étudiants devront ensuite effectuer des requêtes de lecture en utilisant obligatoirement la clé de partition et analyser le comportement du moteur Cassandra. L'objectif est de mettre en évidence les contraintes de requêtage de Cassandra et la nécessité de concevoir le modèle de données en fonction des requêtes.

### Livrable

Une capture ou un compte-rendu montrant la création du schéma, l'insertion des données et l'exécution des requêtes, accompagné d'une courte explication sur le rôle de la clé de partition et sur les différences de fonctionnement entre Cassandra et une base relationnelle.

lien API velib :

<https://opendata.paris.fr/api/explore/v2.1/catalog/datasets/velib-emplacement-des-stations/records?limit=20>