

Redis CLI

Nom Complet : Sghiri Samir

Objectif

L'objectif de ce TP est de vous familiariser avec l'utilisation de Redis pour stocker et manipuler des données dans un contexte réel. Vous allez implémenter une série de tâches courantes qu'une application web pourrait exiger, comme la gestion de sessions utilisateurs, le stockage de messages, la gestion de listes de tâches, et la réalisation de classements.

1. Gestion des Sessions Utilisateurs

Problématique : Stocker et gérer les informations de session des utilisateurs.

Tâches :

1. Créez une clé de session pour un utilisateur avec son ID, contenant ses informations de connexion (nom, email, date de connexion).

```
HMSET session:user:0101 name "sam sgh" email "sam.sgh@example.com" login_date "2024-07-24"
```

2. Mettez à jour l'email de l'utilisateur.

```
HSET session:user:0101 email "sam.sgh001@example.com"
```

3. Récupérez toutes les informations de la session pour cet utilisateur.

```
HGETALL session:user:0101
```

2. Stockage de Messages

Problématique : Stocker et récupérer les messages envoyés par les utilisateurs.

Tâches :

1. Utilisez une liste pour stocker les messages envoyés par un utilisateur.

```
RPUSH messages:user:0101 "Hello, this is my first message!"
```

2. Ajoutez un autre message à la liste.

```
RPUSH messages:user:0101 "This is another message."
```

3. Récupérez les 3 derniers messages pour cet utilisateur.

```
LRANGE messages:user:0101 -3 -1
```

3. Gestion de Listes de Tâches

Problématique : Gérer les listes de tâches des utilisateurs.

Tâches :

1. Créez une liste pour stocker les tâches d'un utilisateur.

```
RPUSH tasks:user:0101 "Tâche 1" " Tâche 2" " Tâche 3"
```

2. Ajoutez plusieurs tâches à la liste.

```
RPUSH tasks:user:0101 " Tâche 4" " Tâche 5"
```

3. Marquez une tâche comme complétée (retirez-la de la liste).

```
LREM tasks:user:0101 1 " Tâche 2"
```

4. Récupérez toutes les tâches restantes pour cet utilisateur.

```
LRANGE tasks:user:0101 0 -1
```

4. Classement des Utilisateurs

Problématique : Gérer un classement des utilisateurs basé sur un score.

Tâches :

1. Utilisez un ensemble trié (sorted set) pour stocker les scores des utilisateurs.

```
ZADD leaderboard 0101 "user:0101" 950 "user:0202" 900 "user:0303"
```

2. Ajoutez des scores pour plusieurs utilisateurs.

```
ZADD leaderboard 850 "user:0404" 800 "user:0505"
```

3. Récupérez le classement des utilisateurs par score.

ZREVRANGE leaderboard 0 -1 WITHSCORES

4. Récupérez la position d'un utilisateur spécifique dans le classement.

ZREVRANK leaderboard "user:0101"

5. Suivi des Présences

Problématique : Suivre quels utilisateurs sont actuellement en ligne.

Tâches :

1. Utilisez un ensemble pour stocker les ID des utilisateurs actuellement en ligne.

SADD online_users 0101

2. Ajoutez un utilisateur à l'ensemble lorsqu'il se connecte.

SADD online_users 1001

3. Retirez un utilisateur de l'ensemble lorsqu'il se déconnecte.

SREM online_users 0101

4. Récupérez la liste des utilisateurs actuellement en ligne.

SMEMBERS online_users

6. Utilisation des Données Géospatiales

Problématique : Permettre aux utilisateurs de partager leur position géographique et de rechercher d'autres utilisateurs à proximité.

Tâches :

1. Utilisez des données géospatiales pour stocker la position des utilisateurs.

GEOADD user_locations 2.3522 48.8566 "user:0101" 3.1390 36.7783 "user:1001"

2. Ajoutez des positions pour plusieurs utilisateurs.

GEOADD user_locations 139.6917 35.6895 "user:0303" -74.0060 40.7128 "user:0404"

3. Recherchez les utilisateurs dans un rayon de 50 km autour d'une position donnée.

GEORADIUS user_locations 2.3522 48.8566 50 km