Building your own social network

We will build a simple social network using Redis as the backend.
We will also see how to design a social graph defining the social interactions or relationships between the users. A social graph is used to show the relationships between the users of your application or website. This is important, as the users are the integral part of a social network.

Instructions

- Work in groups of 2 to 5 people
- Create an API for the main features.
- Each feature can be implemented with your favorite program calling the Redis commands or directly with the Redis commands (categorize them)
- For each creation feature, give the equivalent search feature
- Send me the final result content by e-mail

TIPS

- Use the most appropriate Redis data structure
- Do not forget the usage of Redis transactions

Steps

Step 1: Building the user

User authentication is critical in any social networking application to maintain privacy and to avoid misuse of user accounts.

What information should be stored for a user depends on the kind of social network we want to build. In the case of a personal social network, information such as relationship status, interests, and others are important, whereas in the case of professional networks, educational details, work experience, and skills are more relevant.

For this exercise, we will store basic information such as age, location, gender, and e-mail address, postal address for our users.

Step 2: Building a relationship with the user

In our social network, we will allow people to add other people as friends. However, the friendship request needs to be accepted before the friendship relationship can be established between two users. In order to maintain the friendship, we need to maintain three sets for each user, one for received friend

requests, a second one for all accepted friends, and a third one for all sent friend requests.

Step 3: Adding a friend

To send a new friend request to a user, we will add the user ID of the requesting user into the set.

Step 4: Accepting a friend request

Once the user accepts the friend request, we need to add the user ID to the friends list of both the users.

Step 5: Unfriending a user

Step 6: Updating posts and status

In a real-world social network, depending on the type of network, various kinds of activities such as status updates, photos, links, and videos are allowed. However, for this exercise, we will restrict activity types to posts and status updates only.

Step 7: Status update feed feature

One of the most used features in any social network is the news feed or updates feed, the page containing all the status updates or posts from all the friends. Because we have already stored the updates for any user in the uid: {userid}:updates list, it is very straightforward to query and display the news feed for the user.

Step 8: Commenting on posts feature

Another important feature in any social network is allowing the users to comment
on the posts made by other users. Comments are the most common action taken in a social network. Without comments, the whole social activity on the post does not exist. In this section, we will see what it takes to add a commenting feature to our posts.

Step 9: Focus on the Address part

We want to use data information of the France metropolitan provided by the French data gouv
* Postal code file [1]

- Analyze the files content
- Design the Redis data structure

- Parse the file and store the data into Redis

Sources
[1] - https://www.data.gouv.fr/fr/datasets/correspondance-code-insee-code-postal/

Step 10: Completion on cities and zipCode

Provide an approach of completion for cities and zipCode. Many solutions are available.