Kano User Gifting System Exercise

Social games revolve around virality, and building connections at their core. We would like you to build a simple two way gifting API, for users to interact with one another in a social game environment.

If you are not familiar with the concept, players in Mob Wars: La Cosa Nostra may send gifts to one another once per day, and similarily receive gifts from their friends. This is a nice gesture towards an ally in the game, as these gifts may be used to make your character stronger.

Pre-Requisites

Before beginning, you should complete the following:

- Clone this repository to your workspace.
- Install NodeJS (any version should do).

Notes before you begin

- You may use any NodeJS libraries as part of your submission (we'd recommend [Express](#)).
- We have included `lib/Database.js` and `lib/Collection.js` classes which you may use to simulate a Mongo database environment. The `user` and `gift` collections have been prefilled with data. We have provided you an empty `usergift` collection to work with on this excercise. (These classes could always have issues, please use them if desired, but feel no obligation if they aren't meeting your needs).
- If you would like to work with more data than we have provided, you may add additional data in the `lib/Store.js` file.
- Your submission should not include code that you do not own (open source software, and included libraries are permitted).

Objective

At [Kano](#) we value creativity and problem solving in our employees' day to day work. How you choose to approach and showcase your technical expertise is up to you. While doing so, here are some guidelines for this project:

- You will be building a standalone webserver back-end API in NodeJS, as well as a simple JavaScript/HTML front-end for interacting with your API.
- We would like you to demonstrate your JavaScript ability, as well as data management and server structure chops. Not all candidates, or employees at Kano are familiar with NodeJS, but learning as part of this excercise is perfectly acceptable. **Curiosity** is one of our core values!
- We want you to build a User Gifting API in which users can make the following actions:
  - Users may send a gift to another user on their friends list.
    - `send_gift(senderId: number, recipientId: number, itemId: number)`
  - Users may receive a gift which was sent to them.
    - `receive_gift(recipientId: number, senderId: number)`
  - Users may send a gift to each one of their friends.
    - `send_all(senderId: number)`
  - Users may receive all gifts which have been sent to them.
    - `receive_all(recipientId: number)`
  - Users may fetch a list of their currently pending gifts.
    - `check_gifts(recipientId: number)`

- Users may send one gift to each of their friends **once** per day.

- Users may receive any amount of gifts from their friends each day.

- The server assumes that the client is telling the truth.
  - Simulate being logged in as one of user ID `1`, `2`, `3` or `4`. You could make a dropdown/input field on client side to switch user ID's, or simply manage the "current" user ID through the URL.

Once completed, you should be able to run your API webserver, send it requests, and receive appropriate responses.

Data Spec

- `user` Collection

```
{
  id: number           - The user ID of the user
  img: string          - The user's image URL
  name: string         - The user's name in-game
  friends: Array<number> - An array of user ID's of this user's friends
}
```

- `item` Collection

```
{
  id: number           - The ID of this item
  img: string          - The item's image URL
  name: string         - The name of the item
}
```

Sample Wireframe

Submission Instructions

- Email Magda with the following information:
  - A link to your Github repo, with instructions for us to clone, run and test your submission ourselves.
  - List out any lessons learned from this coding exercise.
  - Would you approach this exercise differently, now that you've completed it?
  - What are some problems that could arise from this implementation?

Questions?

- Email Magda with any questions, issues, or clarifications you may need while working through this.