Full Stack Engineer

Background

For any application with a need to build its own social network, "Friends Management" is a common requirement which usually starts off simple but can grow in complexity depending on the application's use case.

Usually, applications would start with features like "Friend", "Unfriend", "Block", "Receive Updates" etc.

Your Task

Develop an API server that does simple "Friend Management" based on the User Stories below.

You are required to:

- Deploy an instance of the API server on the public cloud or provide a 1-step command to run your API server locally, e.g. using a Makefile or Docker Compose) for us to test run the APIs
- Write sufficient documentation for the APIs and explain your technical choices

User Stories

1. As a user, I need an API to create a friend connection between two email addresses.

The API should receive the following JSON request:

```
{
  friends:
    [
      'andy@example.com',
      'john@example.com'
    ]
}
```

The API should return the following JSON response on success:

```
{
  "success": true
}
```

Please propose JSON responses for any errors that might occur.

2. As a user, I need an API to retrieve the friends list for an email address.

The API should receive the following JSON request:

```
{
  email: 'andy@example.com'
}
```

The API should return the following JSON response on success:

```
{
  "success": true,
  "friends" :
    [
      'john@example.com'
    ],
  "count" : 1
}
```

Please propose JSON responses for any errors that might occur.

3. As a user, I need an API to retrieve the common friends list between two email addresses.

The API should receive the following JSON request:

```
{
  friends:
    [
      'andy@example.com',
      'john@example.com'
    ]
}
```

The API should return the following JSON response on success:

```
{
  "success": true,
  "friends" :
    [
      'common@example.com'
    ],
  "count" : 1
}
```

Please propose JSON responses for any errors that might occur.

4. As a user, I need an API to subscribe to updates from an email address.

Please note that "subscribing to updates" is NOT equivalent to "adding a friend connection".

The API should receive the following JSON request:

```
{
  "requestor": "lisa@example.com",
  "target": "john@example.com"
}
```

The API should return the following JSON response on success:

```
{
  "success": true
}
```

Please propose JSON responses for any errors that might occur.

5. As a user, I need an API to block updates from an email address.

Suppose "andy@example.com" blocks "john@example.com":

- if they are connected as friends, then "andy" will no longer receive notifications from "john"
- if they are not connected as friends, then no new friends connection can be added

The API should receive the following JSON request:

```
{
  "requestor": "andy@example.com",
  "target": "john@example.com"
}
```

The API should return the following JSON response on success:

```
{
  "success": true
}
```

Please propose JSON responses for any errors that might occur.

6. As a user, I need an API to retrieve all email addresses that can receive updates from an email address.

Eligibility for receiving updates from i.e. "john@example.com":

- has not blocked updates from "john@example.com", and
- at least one of the following:
    - has a friend connection with "john@example.com"
    - has subscribed to updates from "john@example.com"
    - has been @mentioned in the update

The API should receive the following JSON request:

```
{
  "sender":  "john@example.com",
  "text": "Hello World! kate@example.com"
}
```

The API should return the following JSON response on success:

```
{
  "success": true
  "recipients":
    [
      "lisa@example.com",
      "kate@example.com"
    ]
}
```

Please propose JSON responses for any errors that might occur.

Constraints

Time

7 days from assignment date. Please feel free to submit your work any time, before the deadline.

Please timebox yourself to a maximum of 4 hours for this activity.

Technology

You are required to use any of these languages: Go, Ruby, JavaScript, Java, PHP or Python.

You are allowed to use any frameworks for the language you chose.

Testing

Please approach this exercise as you would in your day-to-day development workflow.

If you write tests in your daily work, we would love to see them in this exercise too.

Git and Commit History

Sync your app to GitHub and provide the link to the repository to SP rep.
Please maintain a descriptive and clear Git commit history as it would allow us to better understand your thought process.

Follow Up

In the event that you are selected for the next round of interview, please be expected to discuss your work further with us during the interview.

Looking forward to seeing your code!