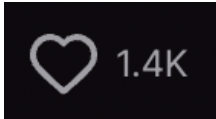


Passes Frontend Take Home

This interview is designed to test frontend engineering basics. Please complete it within one and a half hours. The use of google is allowed. Please read through this entire readme before beginning.

Goal

Our goal is to create a like button like so:



The functionality should be as follows:

- When the button is loaded:
 - If the user has not liked it then show an empty heart icon
 - If the user has liked it then show the heart icon filled with red
 - Display the current number of likes
 - If the count is greater than 999 this amount should be represented in thousands
 - e.g. 1,459 likes should be displayed as 1.4K
- When the button is clicked:
 - If the heart is empty (not liked) then fill the heart with red
 - If the heart is filled (liked) then remove the red from the heart
 - The count displayed next to the button is incremented/decremented

Please create a React component to support these features.

Backend

To facilitate this functionality, we have the following backend APIs:

- GET /api/v1/like/:likeId/count - Returns the number of likes
- GET /api/v1/like/:likeId/user/:userId - Returns a boolean indicating if the user has liked the button
- POST /api/v1/like/add - Adds a like
- POST /api/v1/like/remove - Removes the like

Both POST endpoints expect a request body with fields `likeId` and `userId`.

For testing we have provided a simple server `server.js`. You can run it like so:

```
# in a terminal window
node server.js
```

```
# in another window (example usage)
curl -X POST --data '{"likeId": "123", "userId": "aaron"}' localhost:3001/api/v1/like/add
curl -X POST --data '{"likeId": "123", "userId": "lucy"}' localhost:3001/api/v1/like/add
curl -X GET localhost:3001/api/v1/like/123/count
{"data":2}
```

The server will seamlessly handles new likes and users. No need to "create" a like or user. Also, note that the storage is in memory, so if you spin down the server you will lose any existing data.

Additional Info

Other notes/requirements:

- Use the included `heart-icon.svg` as the heart
- Do not add any dependencies other than `axios` (optional, `fetch` is fine)
- Don't worry about using actually IDs, you can provide arbitrary mock values

Bonuses

These bonus questions are optional. If you would like to complete them and are running out of time feel free to submit the main question and provide a second submission with the bonuses.

Bonus #1

Ensure that if a user rapidly clicks the like button it only results in one API call after they pause clicking for more than 300ms. The UI should be updated optimistically regardless of whether or not the API call has been made.

Bonus #2

Add animation and styling at your discretion. Use the Twitter like button for inspiration.