Startup 06: ADDING A WEB SERVER

Founders:

David Zagreda
Seyedeh Arta Razavi
Colin Stern
SeihakRithy Muth
Brendan Shaughnessy
Jeremy Lee

Search* (*Colin Stern)

Format: GET /api/search [queryText], where [queryText] is a string describing the item being searched for.

Function: This HTTP request returns a list of items that have a title, ISBN or price that matches the query. This request replaces the mock server method getQueryData.

Resources: This HTTP request creates no new resources; it finds and returns items.

Authorization: Any signed-in user can use this request. Since we are allowed to assume that only logged-in users are using the application, there is no check for an authorization token by this request.

User Profile(Jeremy Lee)

Format: GET /user/userid/profile

Function: This HTTP Request gets the profile page of a specific user, and it grabs us the lists of items that a user is interested in purchasing, and that a user is interested in selling. It gets us the user's profile picture, and username.

Resources: This HTTP request returns a profile page for a user containing their profile picture, username, a list of items that they wish to sell, and a list of items that they wish to buy.

Authorization: Any signed-in user can use this request. This will allow them to not only see their personal user profile, but it also allows them to see the user profile of others so that they can see what books people are selling/interested in purchasing.

Home(Seyedeh Arta Razavi)

NOTE: As mentioned below, there are no HTTP routes for this page currently. Instead, we used the Socket API to be able to get home page working for users. We cleared this with Richards to ensure that this was an adequate replacement for implementing an HTTP route.

Create Listing (Brendan Shaughnessy)

Format: POST /api/listing/create

Function: HTTP request takes a json object and adds a new listing and feed items to the database. Replaces mock server functions postItem and postListing.

Resources: New database entries.

Authorization: User needs to be signed in to use this as listings need to be associated with specific users.

Message(David Zagreda)

NOTE: As mentioned below, there are no HTTP routes for this page currently. Instead, we used the Socket API to be able to get messaging working between two users. We cleared this with Richards to ensure that this was an adequate replacement for implementing an HTTP route.

Matched(Rithy)

Format: GET /user/userID/matchedID

Function: Given a matched user id, it will display the listings for both users that they matched upon.

Resources: Displays a list of users you've matched with.

Authorization: Only the signed-in user should be their own matches, outside users will not be able to see this list.

Special Server Startup Procedure:

First, run npm install, and then run npm start. Then, go to localhost:8080 to be able to view our pages.

Individual Contributions

- Colin Stern
    - Search: POST /api/search [queryText] This searches for items which match the queryText in the database.

- Seyedeh Arta Razavi
    - Home: Did not implement an HTTP route, but did use Socket API as a means of loading the home-page and the items for sale by other users and the user's items for sale using the server. (Rithy talked to Richards who said this was a fine alternative instead of using HTTP routes.)

- Jeremy Lee
    - Profile: GET /user/:userid/profile: Essentially will act the same way as the home page will receive listings of items for purchase and items for sale along with the username and user profile picture. There is currently no implementation of the form requests to edit profile information and edit listings, so those buttons have been removed for the time being until they have been implemented.

- David Zagreda
    - Message: Did not implement an HTTP route, but did use Socket API as a means of sending messages using the server.

- Brendan Shaughnessy
    - CreateListing: Adds entries to mongodb.

- SeihakRithy Muth
    - Matched Page and Routes
    - Implemented Backend Server with MongoDB, Socket.IO, and JSON Web Token Authorization

---

Bugs

- Search
    - Only searches for items which match queryText
    - Page selector at bottom of page does not work. Should not be a problem because all results display on one page.

- User Profile
    - Page is a work in progress, we had it working, but currently broken. We are trying to fix it as soon as possible.
- Create Listing
    - Does not interface with mongodb yet.

    - Not fully functional, upon creating listing, it should display on the Homepage, but it currently does not. We are actively working on this, and should have it working soon.
- Message
    - List of sidebar users should be displaying all users that you've matched with so you have the option of messaging these users to discuss buying/selling books. Currently, we just have a mock user that we are assumed to be talking to, but we will try to get this working.
- Matched
    - The listings have disappeared, not too sure why this happened, but working on it.

-

Cuts

- Create Listing
    - We took away features from the items for sale to keep the database from getting too big.

- User Profile
    - Took away the edit profile button for now, will add back to the page once it is implemented.