CS 326: Adding a Database

Due on Wednesday, December 14, 2016

Professor Tim Richards

Daanial Ahmed, Richard Cui, Roman Ganchin,
Gahyun (Susie) Kim, Greg McGrath, Francis Phan

HTTP Requests

GET /items/:itemid

Does: Returns the item with :itemid.
Replaces: getItem(itemId) in the old server
Updates: Does not update the database
Who is authorized: The current user has permission to get all Items in his feed

GET /profile/:userid

Does: Returns the profile for user with :userid.
Replaces: getUserProfile(userId)
Updates: Does not update the database
Who is authorized: Every user has access to their own profile

GET /user/:userid/feed

Does: Returns the feed of the user with id :userid.
Replaces: getFeedData(user, cb)
Updates: Does not update the database
Who is authorized: Every user has access to their own feed

POST /upload/:userid [item]

Summary: Where [item] is a JSON object describing a new item for sale where :userid is the vendor of that item.
Does: Uploads an [item] Replaces: This was a bug in the old submission
Updates: The item list in the database
Who is authorized: Every user has access to upload their own items

PUT/users/:userid/feeds/items/:itemid/like

Summary: Current user has their ID pushed on to the item's likeCounter with id :itemid.
Does: Adds the user's id to the likeCounter
Replaces: likeItem(itemId, userId)
Updates: the likeCounter array for the specified item in the database
Who is authorized: Every logged in user is authorized to like an item

PUT /users/:userid/feeds/items/:itemid/dislike

Does: Adds user id to the dislike counter
Replaces: dislikeItem(itemId, userId)
Updates: the dislike counter array for the specified item in the database
Who is authorized: Every logged in user is authorized to dislike an item

continued on next page. . .

PUT /users/:userid/messages

Does: Sends a message which gets logged by both the sender and receiver
Replaces: New Functionality
Updates: The messages for a particular user
Who is authorized:

GET /user/:userid/pm

Does: Returns the productManager of a user with :userid with all
Replaces: Newly added
Updates: Does not update the database
Who is authorized: Every user is authorized access to their product manager

GET /users/:userid/messages

Does: Returns a list of all the messages for a particular user
Replaces: Newly added
Updates: Messages
Who is authorized: The current user looking at his messages

DELETE /user/:userid/pm/item/:itemid

Summary: Item with :itemid is deleted from user with :userid productManager as it has been sold or removed by user. Also removes the item from all other feeds as it is no longer for sale.
Does: Deletes an item
Replaces: removeItem
Updates: the items that the user owns
Who is authorized: Every user is authorized to remove their own items

POST /resetdb

Summary: Resets database on application.
Does: resets the database
Replaces: Reset server functionality
Updates: To the original database
Who is authorized: Developer feature

Special Server Setup Procedure

We do not have any advanced features that require additional setup.

Individual Contributions

- **Daanial Ahmed** contributed to the Upload component. Contributed to the dynamic functionality of the component (uploading the JSON object to the mock database when the upload button is clicked) which was missing from the previous submission.
- **Richard Cui** contributed to the Chat component. Added functions to retrieve chats from database as well as send chats to other users. Updated database to include support for messages for both senders and recipients. Handled HTTP requests for both functions.
- **Roman Ganchin** contributed to the Trending, Category, Item , Server, and Report components. Added server functionality for dislike item, getCategories, and categorySync. Fixed bug in item.js to enable clicking on images in trending. Changed categories.js so that items could be displayed after they were clicked in feed. Helped writing the report with general outline and filled out the Route components. Updated package.json, and helped move old server functionality into new server.js in server.
- **Gahyun (Susie) Kim** added Bootstrap Warning alert above Cardie layout to display any server issues to client. Installed global function (CardieError) and let code display info in alert box and directed user to try reloading the webpage. Fixed button on uploadItem page to allow the user to select a file. Helped finish writing report.
- **Greg McGrath** contributed to the Feed and item components. Made it so clicking on an item on the homepage no longer goes to item page, but unhides the description that was pre-rendered and hidden. Added server functionality for getItem, and getFeed components. Changed feed.js so that it would contain the functionality for displaying one item at a time.
- **Francis Phan** contributed to the ProductManager and Item components, the dummy database, and linking and routing all of the components into one application. Responsible for implementation of HTTP requests used by ProductManager, Item, Feed, and Upload both from client and server. Refactored ResetDatabase from dummy client-side database to new component that sends HTTP request to reset database on server-side dummy database. Created Schema for Items to be sold.

Lingering Bugs/Issues/Dropped Features

Bugs/Issues

- When clicking an item from trending the functionality works as expected but the url does not change and stays on initial clicked item. Result of changes in item and feed components.
- Chat feature still under development due to complexities with server.