Startup 06: Adding a web server
Team Fig: U-Fig-It-Out

Founders: Craig Fan, Joseph Lew, Gregoire Robinson, T.J. Wilder, Trent Kindvall

HTTP Requests / Routes

-The *verb*, *target*, and *body* of the request.

-What the HTTP request does, and what mock server method it replaces.

-What resources the HTTP request creates/modifies/etc. Examples: "Updates /user/:userid", "Creates a new /user"…

-Who is authorized to use the request

GET /allcategories
Retrieves all categories from the database and passes them to searchleftbar.js. Replaces getCategories(cb). All users are authorized.

GET /categories/:categoryid
Retrieves all items belonging to a category and passes them to searchlistings.js so that these can be rendered in the Search Results page. Replaces getCategoryListings(category, cb). All users are authorized.

GET /user/:userid/info
Retrieves all data pertaining to a particular user id, including username, email, rating, etc. Items that the user is buying or selling is also included in the user info as an array of item ids. GET /items/ can retrieve all items of a user using this array.

GET /items/
Retrieves the information about all the items passed in. Takes a json object containing an item id or a list of such ids and returns a list of item information. This replaces getItemListings(itemIds, cb). All users are authorized.

POST/make_listing/:id
Creates a new listing and adds it to the database using a schema.json file and the method demonstrated in the workshop.

Special Server Setup Procedure

None needed outside of the ordinary

Individual Contributions

TJ: Worked on the /items/ endpoint, updated some code to use better react practices, and fixed and modified a number of small things, mostly on the ItemListing and NewListing pages.

Greg: Worked on getCategories and getCategoryListings and helped Joseph with fixing bugs in searchLeftBar.js and searchListings.js.

Craig: Worked on linking main page categories to search listings. Restructuring project into client and serverside. Migrating database to serverside. Modifying imports on server.js

Joseph: Laid down groundwork for getting the app to work with express routes. Worked on anything the profile page and navbar use. Fixed various bugs and broken links and added the errorbanner for easier debugging.

Trent: Worked on make_listing, created the schema.json object for listing, Adjusted storelisting to incorporate the new database design and changes to make it HTTP compatable.

Lingering Bugs / Issues / Dropped Features

- Make listing can only take small images, should be fixed when we use a real database
- Small issue in prefered payments that won't allow you to select payment option so it cannot be incorporated into itemlisting.json because prefered payments must have a length of one, so xhr cannot stringify the object and we get an error when trying to create a new listing.
- Search bar does not work.
- Main page categories don't work as intended. They don't go to the right place on the search listing page
- The formatting of the item listing page gets messed up when the images are too large and the formatting can change depending on which image is in the primary display