

Startup Submission 06: Adding a Database

Team: BBQForte

Special Server Setup Procedure:

1. Run npm install in /client and /server
2. Run npm run watch in /client
3. Run node src/server.js in /server
4. For Spotify use this free account (I'm not sure how it would work with other accounts until we set up login for the site):
 - a. bbqforte
 - b. cs326
 - c. *This account will be deleted after the class is done, by the way.

Individual Contributions:

Brendan Kelly:

- PUT /user/:userID/name
 - This HTTP request requires a singular string in the body
 - The user sending this request must be the same as userID
 - It is used to edit the full name of the user as seen on the profile page
 - Now uses the database updateOne method
- PUT /user/:userID/about
 - This HTTP request requires a singular string in the body
 - The user sending this request must be the same as userID
 - It is used to edit the about section of a user as seen on the profile page
 - Now uses the database updateOne method

Jess Hendricks:

- Did the initial setup of MongoDB including updating many of the integer _ids to ObjectIDs.
- GET /playlistfeed/user/:userid - the playlist feed updated for MongoDB
 - Updated the getPlaylistFeed() helper function to query the database.
 - Updated the getPlaylist() helper function to query the database and resolve the object ids.
- POST /playlist - a new empty playlist updated for the database
 - Updated the createNewPlaylist() helper function to use the database.
- PUT /playlistfeed/user/:userid/playlist - Adds a playlist from Spotify to the database

- PUT /playlist/:playlistid/songs/:userid - Add a song to a playlist
- DELETE /playlist/:playlistid/spotify/:spotifyauthor/spotifyuser/:userspotify/songs/:songindex/user/:userid - Remove a song from a playlist if that playlist is owned by the user on Spotify
- PUT /playlist/:playlistid - Edit the local information for a playlist (Name, Game, Genre, Description)
- DELETE /playlist/:playlistid - the local copy of the playlist (does not affect the version on Spotify)
- PUT /playlist/:playlistid/votes/:userid - Add a vote(if it doesn't already exists) for the playlist by the current user
- DELETE /playlist/:playlistid/votes/:userid - Remove a vote (if it exists) for the playlist by the current user:

Ka Wo Fong:

- GET /carousel/ - returns the Carousel object stored in database
- GET /news-updates/ - returns the NewsUpdates object stored in database
- GET /most-popular/ - returns the mostPopularData JSON containing the top 8 most popular playlists
- GET /new-release/ - returns the newReleaserData JSON containing the top 8 most popular playlists
- GET /rising/ - returns the risingData JSON containing the top 8 most popular playlists
- GET /highest-rated/ - returns the highestRatedData JSON containing the top 8 most popular playlists

Logan Rennick:

- GET /user/:userID/nickName - returns the nickname of the user of a given userID, useful for the profile page
- GET /forum/ - returns the forum object
- GET /forum/category/:category/topic/:topicID - returns the topic object of a given topicID
 - Now utilizes JSON validation
 - No longer has the "request timed out" error
- POST /forum/category/:category/topic/:topicID/newTopic - Adds a new topic object to the forum object, also updates appropriate thread and post counts
- POST /forum/category/:category/topic/:topicID/thread/:threadID - Adds a new post object to the thread, also updates appropriate post counts
 - Now utilizes JSON validation

Matthew Zenzie:

- PUT '/user/:userID' - allows for modifying user settings and other user information
- PUT '/user/:userID/recommendations/:key' - Used to signal the server to add a recommendation.
- DELETE '/user/:userID/recommendations/:key' - Signal the server to remove the recommendation without adding it.

Stanley Lok:

- GET /private-chat/recent/:userid - Returns the list of users in the 'recent conversations' tab
- POST /private-chat/recent/:userid/add/:otheruserid - Adds a user to the 'recent conversations' tab if that user is not already on the list
- DELETE /private-chat/recent/:userid/remove/:otheruserid - Deletes a user from the 'recent conversations' tab when the 'x' next to the user's name is clicked
- GET /private-chat/live-help/:userid - Returns the list of users to be displayed in the 'live help' tab
 - The user's status (online, away) is displayed beside their name and the users are categorized based on their selected favorite music genre.
- PUT /private-chat/switch/:userid/to/:otheruserid - Switches the chat box to display a conversation with the selected user
 - It does so by updating the 'chattingWith' attribute in the database to the selected user.
- GET /private-chat/conversations/:userid - Returns the main chat box of the private chat and its associated messages
- POST /private-chat/conversations/:userid/create-chat/:otheruserid - Creates a new chat log when the user selected a user whom they have never chatted with
- POST /private-chat/conversations/:userid/chat-with/:otheruseridindex - Sends a message to the other user and displays it in the chat box

Lingering Bugs/Issues/Dropped Features:

1. Spotify playback is only possible at this time with standalone SDKs such as Android or iOS. As such, we have links that will open the Spotify app or website to the playlist in question. (Make sure pop-up block is disabled)
2. The Spotify login is ugly on Chrome. It opens another tab, then after the login it res.redirect('back') to the home page. We would much rather find a way for it to return to the page it was previously on.. Or even better *close* the login window.

3. In the Private Chat, if the user left the page while viewing a chat log that isn't 'Ned Stark's', when the user re-enters the page, Ned Stark's chat log will be displayed instead of what the user recently had it on (this is just a visual bug - clicking on a different user to chat with still functions properly).
4. On the profile page, to access different users' pages, and to see the edited "About" and "Name" sections, you must refresh the page after clicking the link, or clicking save.
5. Dropped favorite game/artist for each user profile
6. There are problems with Steam-related functions. We may end up dropping Steam API integration and just save the game descriptors as attributes to playlists.
7. The Rising Playlist breaks the server (causes connection to the server to break).
8. In the settings page, if you click on the tick box for "Public Profile," you will receive an error message.