

How to get data from backend:

https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch

<https://stackoverflow.com/questions/45535913/setting-authorization-header-in-fetch-api>

<https://remotestack.io/react-local-storage-tutorial-store-form-state-in-localstorage>

Career Cluster Routes

These routes are used to get data about clusters, pathways, and careers.

You can find the data under `server/static/careerclusters.js`

`GET /api/clusters/get_clusters`

Returns a list of clusters

`GET /api/clusters/get_cluster_short/:name`

Returns a single cluster. The name parameter is the short name of the cluster.

`GET /api/clusters/get_pathways`

Returns a list of pathways

`GET /api/clusters/get_pathways_via_cluster/:cluster`

Returns a list of pathways under a single cluster. The cluster parameter is the short name of the cluster.

Career Routes

These routes are used to get data about careers and like them.

GET /api/career/get_careers

Returns a list of careers. If logged in, will also include whether that career is liked or not. If not logged in, the liked variable will always be false.

GET /api/career/get_careers_via_pathways/:pathway

Returns a list of careers that falls under a particular pathway. The pathway parameter uses the short name. Has the same login behavior as the get_careers route.

GET /api/career/get_career/:id

Returns a single career given its career id. The id can be obtained from the previous two routes. Has the same login behavior as the get_careers route.

POST /api/career/create_career

Creates a career and returns it. Name, salary, and pathway are required. Description and image are optional. Pathway uses the short name.

POST /api/career/update_career/:id

Updates a career given its id and returns the updated value. Requires a login.

DELETE /api/career/delete_career/:id

Deletes a career given its id and returns the deleted career. Requires a login.

GET /api/career/like_career/:id/:liked

Likes/unlikes a career based off of its id, based off of the liked parameter. Returns the updated career. Requires the user to be logged in. Returns a 400 if the career wasn't found or if the user isn't logged in.

GET /api/career/get_liked_careers

Returns a list of careers the user has liked. Requires a login.

User Routes

These routes are used to get information about users and to login/register.

POST /api/user/register

Used to register a new user. Requires a JSON at the body containing a name, email, and password. Returns a 400 upon an error and a 200 if no errors were encountered. Also returns the user with their hashed password if properly registered.

POST /api/user/login

Used to login. Requires a JSON at the body containing the email and password. Returns a 400 if login is incorrect, and a 200 otherwise. Returns the user's JWT token, ID, and isAdmin upon successful login (use it to access routes requiring login).

User (non-admin) email: jeremytinana@gmail.com

User (non-admin) password: bCryptSync

Admin email: imadmin@gmail.com

Admin password: GibPermsPLZ

POST /api/user/update_self

Updates their own passwords and stuff. Cannot self-elevate. Requires a login

DELETE /api/user/delete_self

Deletes their own profile. Requires a login.

POST /api/user/admin_update/:user

Updates a user's information. Can elevate users. Requires an admin login

DELETE /api/user/admin_delete/:user

Deletes a user profile. Requires an admin login.

GET /api/user/get_users

Returns the list of users. The contents only contain their name, signature, pfp, and sport.

GET /api/user/search_users/:search

Returns the list of users with a matching name or email. Contents are the same as the previous route. Behavior may be buggy, try using the previous route if possible.

GET /api/user/get_user/:id

Returns a user based off of their user id. Contents are the same as the previous route.

GET /api/user/get_self_profile

Returns the user's profile. Requires a login. Contents are the name, pfp, signature, email, and sport.

GET /api/user/get_self_mindmap

Returns the user's top 3 (by number) clusters. Requires a login. If 1st place has a three-way tie, will return 5 clusters. Returns data in correct format for d3 module.

Quiz Routes

These routes are used to get information about quizzes

GET /api/quiz/get_quizzes

Returns a list of possible quizzes

GET /api/quiz/get_quiz/:id

Returns a single quiz given its quiz id. You can find the quiz id in the get_quizzes route

POST /api/quiz/create_quiz

Creates a new quiz. Requires login with admin perms

POST /api/quiz/update_quiz/:id

Updates a single quiz given its quiz id. Requires login with admin perms. Removes all instances of that saved quiz.

DELETE /api/quiz/delete_quiz/:id

Deletes a single quiz given its quiz id. Requires login with admin perms. Removes all instances of that saved quiz.

GET /api/quiz/get_saved/quizzes

Returns the user's saved quizzes. Requires a login.

GET /api/quiz/get_saved_quiz/:id

Returns a singular saved quiz the user has. Requires a login

POST /api/quiz/save_quiz

Creates a new saved quiz and updates the user's mindmap. Requires a login. Send the quizID and answers (array of answerIDs). Cluster data will automatically be taken care of.

POST /api/quiz/update_saved_quiz/:id

Updates a previously saved quiz and also updates the user's mindmap. Requires a login. Send the answers and points.

DELETE /api/quiz/delete_saved_quiz/:id

Deletes a previously saved quiz and also updates the user's mindmap. Requires a login