Backend Documentation

Each backend file (endpoint) has been highlighted in yellow below. Please read the information associated with each one before using them within development.

app.py

- File contains the Flask App Structure. (*consider as an index file*)
- Connects all the endpoints (*other .py scripts*) into one environment.

__pycache__

- Holds cache files for python Flask development. (ignore, can be deleted but preferred to keep).

Tests folder

- Contains the necessary supporting backend unittests.
- These tests help ensure that the application is running how it was intended to. These tests may need to be updated if further development is done in the future.

__init__.py

- Blank script, needed to structure the flask environment (*ignore*).

auth.py [POST]

- Endpoint handles the user sign-in/sign-out process.
- Inputs for /**signout:** {authkey}; Returns False if success logout, True if error during logging out.
- Inputs for /**signin**: {email, password}; Returns {authkey, status (of login-state/error-message)}

client.py

- Acts as client file for connecting to database.
- Include in all files that require database connection or require the MariaDB connector.

createAccount.py [POST]

- Endpoint creates a new user account within the database.
- Inputs: {email, first_name, last_name, password, isClient}
- Returns {authkey}; authkey = null if unable to create new account.

deleteAccount.py [POST]

- Delete user surveys & answers, and other data from database.
- Inputs: {authkey}
- Returns {status, authkey}; Status notifies frontend of success/not for deleting account.

disableQuestion.py [POST]

- Disable a given question (qid), Only admin accounts.
- Inputs: {authkey, qid}
- Returns {status}; status = 'success'/'fail'.

enableQuestion.py [POST]

- Enable a given question (qid), Only admin accounts.
- Inputs: {authkey, qid}
- Returns {status}; status = 'success'/'fail'.

editInfo.py [POST]

- Update/edit account information for a given account (*authkey*).
- Inputs: {authkey, first_name, last_name, email, client}
- Returns {uid}; Returns uid for account that information was updated for.

editPassword.py [POST]

- Update/change account password.
- Inputs: {authkey, current_pass, new_pass}
- Returns {status}, Status of account password update operation.

getAccount.py [POST]

- Fetch account information for given account from users, and preferences database tables.
- Inputs: {authkey}
- Returns {email, first_name, last_name, date_created, last_login, isClient, isAdmin, circle_colors, circle_rank, notify_time, avatar}

editInfo.py [POST]

- Update/edit account information for a given account (*authkey*).
- Inputs: {authkey, first_name, last_name, email, client}
- Returns {uid}; Returns uid for account that information was updated for.

getQuestions.py [GET, POST]

- Fetch array of questions stored within the database.
- Inputs: {getDisabled}; Fetch questions that are disabled (1) or not (0).
- Returns {questions[]};

getScore.py [POST]

- Get score for a given account.
- Inputs: {authkey}
- Returns {scores}; Json containing information about scores.

getUsers.py [GET]

- Fetch list of users from users' table. Only available to admin accounts.
- Inputs: {authkey}
- Returns {users}; Returns uid for account that information was updated for. (*null if non-admin account*)

preferences.py [GET, POST]

- Updates notify_time, domain colors, domain ranks, avatar settings for a given account.
- Inputs: {authkey, setting, value}; Setting is one of the four settings listed above. Value is the new value to set it to.
- Returns {uid, setting, value}; Returns uid for updated account, and the setting that the updated, plus the value set.

removeQuestion.py [GET, POST]

- Remove a question from the database.
- Inputs: {authkey, qid}
- Returns {status}; Status = true (*deleted*) / false (*error in deleting process*).

makeAdmin.py [POST]

- Allows admin to set other accounts as administrators. (*Give admin privileges*).
- Inputs: {authkey, email, makeAdmin}
- Returns {status}; 'success' / 'fail'

submitAnswers.py [POST]

- Submit answers for a given user.
- Inputs: {answers, authkey}
- Returns {message}; Status of whether answers were saved into database or not.

updatePrefs.py [POST]

- Shortcut method of updating all three (*notify_time, circle_colors, circle_rank*) as (pass1, pass2, pass3) respectively.
- Inputs: {authkey, pass1, pass2, pass3}
- Returns {uid}; Returns uid for account that information was updated for.

addQuestion.py [POST]

- Insert list of questions into the questions table; Available to admin accounts only.
- Inputs: {authkey, questions[]}
- Returns {qids[]}; Returns list of qids for the newly created questions.

editQuestion.py [POST]

- Edit a given question in the questions table; Available to admin accounts only.
- Inputs: {authkey, qid, type, title, weight, disabled, subQuestions[]}.
- Returns {status}; true (*question updated*) / false.