API Endpoints Outline

Auth Routes

API Endpoint = POST => “api/v1/auth/register” => Success Message
A successful response will respond with 201 and create the session cookies with the Access and Refresh Tokens. A verification will also be sent to the email the user used to register.
An error response will respond with 400 and return register errors.
An error response will respond with 409 conflict if email is already in use.

API Endpoint = POST => “api/v1/auth/login” => Success Message
A successful response will respond with 200 and create the session cookies with the Access and Refresh Tokens. Google reCaptcha service is used to prevent bot traffic.
An error response will respond with 400 Bad Request if input is malformed.
An error response will respond with 401 Unauthorized if invalid credentials used.
An error response will respond with 429 Too Many Requests to rate limit for brute-force protection.

API Endpoint = POST => “api/v1/auth/logout” => Success Message and Revoke Refresh Token
A successful response will respond with 200 and delete the client cookies and revoke validity of only the Refresh Token. Since Access Tokens are short lived, they are not revoked.
An error response will respond with 500 and return logout errors.

API Endpoint = POST => “api/v1/auth/refresh” => Success Message
A successful response will respond with 201 and issue a new access token + refresh token to the user. The refresh token and the userUUID in the token payload must match the token payload stored in the server redis cache. The refresh token used to generate new tokens will either be discarded or get cached in the redis cache as a revoked token with its remaining lifetime.
An error response will respond with 401 Unauthorized if the refresh token has been revoked.
An error response can also respond with 409 if the client provided userUUID doesn’t match the userUUID for the refresh token stored in the server redis cache.

User Routes A userUUID hashing function creates user public identifiers. Keeps actual uuids safe. NOTE: Three unique identifiers are in use, each having their own responsibilities.

-__id: mongodb unique identifier. Created by mongodb and used by mongodb only.
-UUID: Created by server using uuidv4. This identifier is used by the server auth functions to look up a user and is stored in the jwt accessToken payload and refreshToken redis cache.
-publicId: 9 character string created by server using the previous UUID, crypto, and basex functions. Displayed to other users and is only used in identifying users. Has no significance to server auth.

API Endpoint = GET => “api/v1/users/me” => Private User Profile Information
A successful response will respond with 200 and provide the username and publicId for the logged in user. This endpoint will never respond with an error status, but it will succeed and not return any information if an error occurs.

API Endpoint = POST => “api/v1/users/me” (unused) => Updated Private User Profile Information
A successful response will respond with 200 and provide the updated private profile information for the logged in user. Will never display anything other than the current user’s updated private information.
An error response will respond with 401 Unauthorized if the user's access token has expired.

API Endpoint = GET => “api/v1/users/:userUUID” (unused) => Public User Profile Information
A successful response will respond with 200 and provide the public/private user profile information depending on the accessToken userUUID.
An error response will respond with 401 Unauthorized if the user's access token has expired.
Another error response will respond with 404 if the userUUID doesn’t exist.

API Endpoint = GET => “api/v1/users/:userUUID/workouts” => User Created Workouts
A successful response will respond with 200 and show all workouts the specified userUUID has created.
Another successful response may respond with 200 and an empty list if the user has not created any workouts
An error response will respond with 401 Unauthorized if the user's access token has expired.
An error response will respond with 404 if the userUUID doesn’t exist.

API Endpoint = GET => “api/v1/users/:userUUID/meals” => User Created Meals
A successful response will respond with 200 and show all meals the specified userUUID has created.
Another successful response may respond with 200 and an empty list if the user has not created any meals.
An error response will respond with 401 Unauthorized if the user's access token has expired.
An error response will respond with 404 if the userUUID doesn’t exist.

API Endpoint = POST => “api/v1/users/:userUUID/reports” => reports data
A successful response will respond with 201 and the data points to be used by the frontend to generate progress charts.
Another successful response may respond with 200 and an empty list of data if no user data has been recorded.
An error response will respond with 400 Bad Request if invalid filter params are provided.
An error response will respond with 401 Unauthorized if the user's access token has expired or if the access token userUUID doesn’t match the userUUID sent as a url parameter.
An error response will respond with 404 if the userUUID doesn’t exist.

Exercise Route

API Endpoint = GET => “api/v1/exercises” => All Exercises
A successful response will respond with 200 and provide the information for all exercises.
An error response will respond with 401 Unauthorized if the user's access token has expired.

API Endpoint = GET => “api/v1/exercises/:exerciseID” => Specified Exercise Information
A successful response will respond with 200 and provide the information for the specified exercise.
An error response will respond with 401 Unauthorized if the user's access token has expired.
Another error response could respond with 404 if the specified exerciseID doesn’t exist.

Workouts Route

API Endpoint = POST => “api/v1/workouts” => Confirmation of User Created Workout
A successful response will respond with 201 and provide a confirmation message with the information for the user created workouts.
An error response will respond with 400 Bad Request if a validation error occurs.
An error response will respond with 401 Unauthorized if the user's access token has expired.

API Endpoint = GET => “api/v1/workouts” => All User Created Workouts
A successful response will respond with 200 and provide the information for all user created workouts.
A successful response could also respond with 200 and provide an empty list if no user created workouts exist.
An error response will respond with 401 Unauthorized if the user's access token has expired.

API Endpoint = GET => “api/v1/workouts/:workoutID” => User Created Workout
A successful response will respond with 200 and provide the information for the specified user created workout.
An error response will respond with 400 Bad Request if a validation error occurs.
An error response will respond with 401 Unauthorized if the user's access token has expired.
Another error response could respond with 404 if the specified workoutID doesn’t exist.

API Endpoint = DELETE => “api/v1/workouts/:workoutID” => User Created Workout
A successful response will respond with 200 and delete the specified user workout.
An error response will respond with 401 Unauthorized if the user's access token has expired or if the owner userUUID doesn’t match the requesting userUUID.
An error response could respond with 404 if the specified workoutID doesn’t exist.

Meals Route

API Endpoint = POST => “api/v1/meals” => Creates a Meal
A successful response will respond with 201 and provide a successful meal creation message.
An error response will respond with 400 Bad Request if a validation error occurs.
An error response will respond with 401 Unauthorized if the user's access token has expired.

API Endpoint = GET => “api/v1/meals” => All User Created Meals
A successful response will respond with 200 and provide the information for all user created meals.
A successful response could also respond with 200 and provide an empty list if no user created meals exist.
An error response will respond with 401 Unauthorized if the user's access token has expired.

API Endpoint = GET => “api/v1/meals/:mealID” => User Created Meal
A successful response will respond with 200 and provide the information for the specified user created meal.
An error response will respond with 401 Unauthorized if the user's access token has expired.
Another error response could respond with 404 if the specified mealID doesn’t exist.

API Endpoint = DELETE => “api/v1/meals/:mealID” => Successful Deletion Message
A successful response will respond with 200 and delete the specified user meal.
An error response will respond with 401 Unauthorized if the user's access token has expired or if the owner userUUID doesn’t match the requesting userUUID.
An error response could respond with 404 if the specified mealID doesn’t exist.

*Most endpoints may respond with 500 IF a “general server error occurs”