

HTTP requests should be made to *http://localhost:5000*

CONNECTION TO API

Register User

A user must be registered to utilize the API. To register a user, send POST to *http://localhost:5000/users/register* with *email* and *password* fields in the POST body.

Login

Once the user has registered, you will need to send another POST to *http://localhost:5000/users/login* with *email* and *password* fields in the POST body. If the login is successful, the server will return with a session cookie as well as user information (*user_id*, *email*, *isAdmin*) in the body of the response.

USER MANAGEMENT

Admin only functions

GET user

GET request to *http://localhost:5000/users* will query database for user information.

DELETE user

DELETE request to *http://localhost:5000/users?user_id=<id>* will remove the desired user.

REGISTER user

POST request to *http://localhost:5000/users/register* will create a new user. The *email* and *password* fields must be defined in the HTTP body. If the desired user is to be an administrative user, then include *isAdmin=true* in the body as well.

PATCH (Admins)

PATCH request to *http://localhost:5000/users?user_id=<id>* will modify attributes of a given user. If no *user_id* is specified, then the requesting user is assumed. The modified attributes of the user must be present in the HTTP body. If successful, the new user object will be returned.

**Admins can set 'isAdmin=false' for other admins, but they are unable to do so for themselves.*

Standard functions

Login

POST request to *http://localhost:5000/users/login* with *email* and *password* fields defined in the HTTP body. Upon successful login, a cookie will be stored on the client browser with the session details. This cookie will expire after 24 hours.

PATCH (Non-admin)

PATCH request to *http://localhost:5000/users* will modify the current user's attributes (excluding isAdmin). The attributes to be modified must be present in the HTTP body. If successful, the new user object will be returned.

API FUNCTIONS

There are two resources to make changes to: donations and donators

(*http://localhost:5000/donations* or *http://localhost:5000/donators*)

GET

Just a basic GET request will query the table for all instances of a resource

To request a resource based on a specific property or value, then append the query operator *?* and *resource_name=value* key/pairs can be stringed together with *&*

Ex/ *http://localhost:5000/donations?item_type=chair&donator_id=11*

Distinct column values

To retrieve all distinct column values of a given attribute, use

http://localhost:5000/donations/column?attribute=<attribute>

This will return an array of distinct values.

Dates

Specific dates, range of dates, and absence of dates can be specified. The date format is *yyyy-mm-dd*.

Specific date: *http://localhost:5000/donations?date_donated=2020-04-03*

Date range: *http://localhost:5000/donations?date_donated_start=2020-04-02&date_donated_end=2020-04-05*

* if a start or end value does not have to be specified

If you want to return all resources that have not been donated, then you can use *?date_donated=null*.

POST

POST requests are used to add new data to the database. POST relies on the body of the http request to have the necessary information to add to the database. The body should basically just be a series of key value pairs (in our case, we will be using JSON). All POST requests should be made to the appropriate URI. If you want to add data to the donators table, you will use */donators* and if you want to add data to the donations table, use */donations*.

A successful POST will return a JSON of the data added to the database (with a generated id).

*NOTE – when POSTING or PATCHING, be sure to use x-www-form-urlencoded format

PATCH

PATCH requests are used to modify existing data in the database. In order to use PATCH, you will need to specify your resource with the id value of the object using the query operator. So if you want to make a modification a donation, you will use */donations?donation_id=x*. To make a change to a donator, use */donators?donator_id=x*.

In the body of the http request will be the key/value pairs of the data you want changed.

PATCH will return a JSON of the modified data.

*NOTE – when POSTING or PATCHING, be sure to use x-www-form-urlencoded format

DELETE

DELETE requests will remove a resource from the database. To delete a resource, specify the id. If you want to remove a donation, you will use */donations?donation_id=x*. To remove a donator, use */donators?donator_id=x*.

TABLE ATTRIBUTES

Users

user_id (unsigned int)

email (string)

password (string)

isAdmin (Boolean)

Donations

donation_id (unsigned int)

donator_id (unsigned int)
item_tag (string)
item_condition ('new', 'great', 'good', 'poor')
item_type (string)
item_quantity (unsigned small int)
date_received (date)
date_donated (date)
donate_name (string)
donate_town (string)

Donators

donator_id (unsigned int)
donator_name (string)
donator_email (string)