

APP URL: <http://final15.appspot.com>

NOTE: OBTAIN YOUR JWT VIA BROWSER THEN PLEASE TEST API USING POSTMAN NOT THE BROWSER. Thank You!

(Also note I do not provide user_id in the environment file for postman since the user_id is stored within the jwt once you logged in)

ROUTE GET /welcome

DESCRIPTION: This is the directory where it shows you your jwt and username

HEADERS REQUIRED:
NA

SUPPORTED STATUSES:

200

VALID API CALLS

(No api calls needed since this is a redirection performed by the google api once you login on the "/" route)

ROUTE GET /

DESCRIPTION: This is where you can login to google

HEADERS REQUIRED:
NA

SUPPORTED STATUSES:

200

VALID API CALLS

(No api calls needed since you can do this on the browser)

ROUTE: GET *

DESCRIPTION: In this route if you specify anything that does not consist of the format mentioned on the routes below then it will return a status of 405 where the method is now allowed

HEADERS REQUIRED:
NA

SUPPORTER STATUSES

405

VALID (invalid) API CALLS

get boogers/284728

get boats/284728/boats

ROUTE: GET /boats OR /boats?page=1&limit=5

DESCRIPTION: In this route if you are logged in then it will display both you public and private boats along with other users boats but they must be public if they are not they will not be shown.

HEADERS REQUIRED:

Content-Type: application/json

Authorization: Bearer [Token Here] (OR WITHOUT)

SUPPORTED STATUSES:

200

401

VALID API CALLS

get /boat (with jwt)

get /boat (no jwt)

ROUTE GET /owners/:username/boats

DESCRIPTION: In this route you may request to see boats from a specific user but you are not allowed to see public boats if you are not authenticated but if you are logged in and you are the user added in the url then you will be able to see your private boats

HEADERS REQUIRED:

Content-Type: application/json

Authorization: Bearer [Token Here] (OR WITHOUT)

SUPPORTED STATUSES:

200

400

401

403

VALID API CALLS

get owners/:username/boat (with jwt and is user)

get owners/:username/boat (with jwt and is not user)

get owners/:username/boat (no jwt)

ROUTE GET /boats/:id

DESCRIPTION: In this route you may request a specific boat but if the boat is private then you are not able to view it with login or without login. If the boat is public a non logged in user may see it or a logged in user may see it. If you are logged in and you are the user that owns that boat then you will be able to see it even if it's a private boat.

HEADERS REQUIRED:

Content-Type: application/json

Authorization: Bearer [Token Here] (OR WITHOUT)

SUPPORTED STATUSES:

200

400

401

404

VALID API CALLS

get boats/:id (with jwt and is user)

get boats/:id (with jwt and is not user)

get boats/:id (no jwt)

ROUTE POST /boats

DESCRIPTION: In this route you may be able to post but you **MUST** be logged in in order to perform the request.

HEADERS REQUIRED:

Content-Type: application/json

Authorization: Bearer [Token Here] (OR WITHOUT)

SUPPORTED STATUSES:

201

401

405

406

VALID API CALLS

get boats/:id (with jwt and is user)

get boats/:id (with jwt and is not user)

get boats/:id (no jwt)

ROUTE DELETE /boats/:id

DESCRIPTION: In this route you may be able to delete your boats which in turn will disassociate any loads that are in the boat (but it will not delete the load). In order to properly perform this request you must be the user who owns the boat and must be logged in and the boat must be valid. If you are not logged in then you will not be permitted to use this route.

HEADERS REQUIRED:

Content-Type: application/json

Authorization: Bearer [Token Here]

SUPPORTED STATUSES:

201

401

406

VALID API CALLS

Delete boats/:id (with jwt and is user)

ROUTE POST /loads

DESCRIPTION: In this route anyone can post a load unless you are not logged in. The format for sending a load must be in json format.

HEADERS REQUIRED:

Content-Type: application/json

Authorization: Bearer [Token Here]

SUPPORTED STATUSES:

201

401

405

406

VALID API CALLS

post boats (with jwt)

ROUTE: GET /loads OR /loads?page=1&limit=5

DESCRIPTION: In this route you are able to get all loads. This can be with your jwt or without.

HEADERS REQUIRED:

Content-Type: application/json

Authorization: Bearer [Token Here] (OR WITHOUT)

SUPPORTED STATUSES:

200

VALID API CALLS

get /load (with jwt)

get /load (no jwt)

ROUTE GET /loads/:id

DESCRIPTION: Loads will always be public but can have a carrier. Carriers will be public even if the boat is private. You do not need a jwt token in order to perform the request but you can have it if you want to

HEADERS REQUIRED:

Content-Type: application/json

Authorization: Bearer [Token Here] (OR WITHOUT)

SUPPORTED STATUSES:

200

400

401

404

VALID API CALLS

get loads/:id (with jwt)

get loads/:id (no jwt)

ROUTE DELETE /loads/:id

DESCRIPTION: In this route you can delete a load but if a load is associated with a boat that you do not own then the method will not be allowed. If the load is not associated with a boat then any user can delete it so long as they are logged in. You may be able to delete a load that is associated with a boat but in order to do that you need to own that boat that carries that load.

HEADERS REQUIRED:

Content-Type: application/json

Authorization: Bearer [Token Here]

SUPPORTED STATUSES:

200

400

401

404

VALID API CALLS

Delete loads/:id (with jwt is user)

Delete loads/:id (with jwt is not user)

ROUTE GET /users

DESCRIPTION: Get all the user that are registered for the application

HEADERS REQUIRED:

Content-Type: application/json

Authorization: Bearer [Token Here] (OR WITHOUT)

SUPPORTED STATUSES:

200

VALID API CALLS

Get /users

ROUTE PUT /boats/:bid/loads/:lid

DESCRIPTION: In this route you can add a load to a boat. The id must exist before being able to do the query. If the load is already assigned

HEADERS REQUIRED:

Content-Type: application/json

Authorization: Bearer [Token Here]

SUPPORTED STATUSES:

400

404

403

204

VALID API CALLS

PUT /boats/:bid/loads/:lid (with jwt)