Final Project

Matthew Egan

CS 493: Cloud Application Development

Fall 2019

Oregon State University

Last Update: December 9, 2019. 8:00 pm Central

URL application deployed on GCP: https://eganmat-final-261504.appspot.com

URL to login and get account created for application: https://eganmat-final-261504.appspot.com

Non-User entities, Boats and Loads:

1. Boats: A boat can have only one owner(a user that has been added to the database). A boat can have 0, 1, or many loads added to it. When an authenticated user creates a boat, the 'sub' property of the user's JWT is used as the foreign key in a boat's 'owner' property. Only

the user that created the boat can then edit(with either PUT or PATCH) or delete that boat. The owner of that boat is also the only one allowed to add or remove loads to that specific boat. When a user adds a load to their boat, the load's datastore 'id' is added to a 'loads' array as a foreign key. When a load is removed from the boat, or the load is deleted, the boat's 'loads' array is updated by removing that loads id.

2. Loads: A load can have only one boat it can be put on. When a load is created, it has a property 'carrier' that is set to Null. When a user adds a load to one of their boats, the 'carrier' property of that load is updated with the datastore 'id' of the boat it was added to. When a boat is deleted or a load is removed from a boat, the load's 'carrier' property is updated to be Null.

Users and authentication:

1. When users login to the application, they are added to the database of the application by storing the 'sub' property of the JWT as a unique identifier, along with their name and email. Only users who have been added to the database can make protected requests. When an authorized and authenticated user makes protected requests to the application, a valid JWT must be provided.

Protected requests:

1. POST /boats
    a. User must be in database
2. PUT /boats/boat_id
    a. User must be in database
    b. User must be owner of boat provided
3. PATCH /boats/boat_id
    a. User must be in database
    b. User must be owner of boat provided
4. DELETE /boats/boat_id
    a. User must be in database
    b. User must be owner of boat provided
5. POST /loads
    a. User must be in database

6. PUT /loads/load_id
    a. User must be in database
7. PATCH /loads/load_id
    a. User must be in database
8. DELETE /loads/load_id
    a. User must be in database
9. PUT /boats/boat_id/loads/load_id
    a. User must be in database
    b. User must be owner of boat provided
10. DELETE /boats/boat_id/loads/load_id
    a. User must be in database
    b. User must be owner of boat provided

Notes on viewing relationship of a user and boats:

1. Users can have zero, one, or many boats. In order to view all boats of a single user, a GET request is sent to GET /boats, with a JWT in the authorization header as a bearer token. This will return a list of any boats that the user has created. If a user sends a GET request to GET /boats without a JWT in the Authorization header, it will return a list of all boats of every user.

Notes on Postman testing, PLEASE READ:

In order to run my Postman tests, and test the application, the tester needs to obtain a JWT using the login for this application from 2 different Google accounts, and enter those JWTs into token and token2 in the Postman environment variables.

Important:

I also have a requirement that when a user provides a JWT in a protected request, the user has to be a member of my application. In order to test this in my Postman collection, the tester needs to get a 3rd JWT token from a third Google account. I still have my assignment 7 running, so the tester can visit this URL:

https://eganmat-hw7-authent-auth.appspot.com/

and login with a Google account not used for the final project application to get JWT that is valid but that account is not added to the database. In the Postman environment variables, this will be token3. IMPORTANT REMINDER: for the tests to run correctly, the account used in this homework 7 URL, it needs to be an account that hasn't been used for the final project application.

Any questions, please email me at eganmat@oregonstate.edu

Create a Boat

Allows you to create a new boat.

POST /boats

Request

Authentication requirement:

Required. A valid JWT with the user being a part of the database

Request Parameters

None

Request Body

Required

The request body must contain all three attributes: name, type, length

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
name	String	The name of the boat.	Yes
type	String	The type of the boat. E.g., Sailboat, Catamaran, etc.	Yes
length	Integer	Length of the boat in feet.	Yes

Request Body Example

```
{
  "name": "Sea Witch",
   "type": "Catamaran",
  "length": 28}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, the boat must not be created, and 400 status code must be returned.
Failure	401 Unauthorized	No JWT provided, 401 returned JWT valid, but user not in database, 401 returned

		JWT invalid, 401 returned
Failure	406 Not acceptable	If application/json not listed in Accept header

Response Examples

Success

```
Status: 201 Created

{
  “id”: “abc123”,
  “name”: “Sea Witch”,
  “type”: “Catamaran”,
  “owner”: user_id
   “loads”: []
  “length”: 28,
  “self”: “https://<your-app>/boats/abc123”
}
```

Failure

```
Status: 400 Bad Request

{
“Error”:  “The request object is missing at least one of the required attributes”
}
```

Failure

```
Status: 400 Bad Request

{
“Error”:  “The body of the request is missing”
}
```

Failure

```
Status: 401 Valid JWT but user not in database
{
“Error”:
Invalid user.  You are not in the database.  Please visit homepage to login
}
```

Failure

```
Status: 401 JWT missing
{
“Error”:  JWT missing.  You are not authorized to make that request
}
```

Failure

```
Status: 401 JWT invalid
{  “Error”:  Invalid JWT.  You are not authorized to make that request
}
```

Failure

Status: 406 Not Acceptable

```
{
“Error”: “The requested MIMEtype is not allowed”
}
```

Get a Boat

Allows you to get an existing boat

```
GET /boats/:boat_id
```

Request

Request Parameters

Name	**Type**	**Description**	**Required?**
Boat_id	String	ID of the boat	Yes

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	**Status Code**	**Notes**
Success	200 OK	
Failure	404 Not Found	No boat with this boat_id exists
Failure	406 Not Acceptable	If application/json not listed in accept header

Response Examples

Success

```
Status: 200 OK

{
  "id": "abc123",
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "self": "https://<your-app>/boats/abc123"
}
```

Failure

```
Status: 404 Not Found

{
"Error":  "No boat with this boat_id exists"
}
```

Failure

```
Status: 406 Not Acceptable

{
“Error”:  “The requested MIMEtype is not allowed”
}
```

View all Boats

List all the boats.

GET /boats

Request

Authorization required:

Depends. If a user provides a JWT in the Authorization header,

Request Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	If application/json not listed in accept header

Response Examples

Success

```
Status: 200 OK

[
{
  "id": "abc123",
  "name": "Sea Witch",
   "type": "Catamaran",
  "length": 28,
  “loads”: []
  "self": "https://<your-app>/boats/abc123"
},
{
  "id": "def456",
  "name": "Adventure",
   "type": "Sailboat",
  "length": 50,
  “loads”: []
  "self": "https://<your-app>/boats/def456"
},
{
  "id": "xyz123",
  "name": "Hocus Pocus",
   "type": "Sailboat",
  "length": 100,
  “loads”: []

  "self": "https://<your-app>/boats/xyz123"
}
]
```

Failure

```
Status: 406 Not Acceptable


{
“Error”:  “The requested MIMEtype is not allowed”
}
```

Delete a Boat

Allows you to delete a boat. Note that if the boat is currently in a slip, deleting the boat makes the slip empty.

```
DELETE /boats/:boat_id
```

Request

Authorization required:

JWT must be provided, and user must be in the database

Request Parameters

None

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	JWT provided, user not in database Invalid JWT No JWT
Failure	403 Forbidden	JWT provided and user in database, but not owner of boat to be deleted
Failure	404 Not Found	No boat with this boat_id exists

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 404 Not Found

{
"Error":  "No boat with this boat_id exists"
}
```

Failure

Status: 403 Forbidden

```
{
“Error”:
“The boat you selected is owned by someone else.  You are not authorized to make th
at request”
}
```

Failure

Status: 401 Valid JWT but user not in database

```
{
“Error”: {
Invalid user.  You are not in the database.  Please visit homepage to login
}
}
```

Failure

Status: 401 JWT invalid

```
{  “Error”:  Invalid JWT.  You are not authorized to make that request
}
```

Failure

Status: 401 JWT missing

```
{
“Error”:  JWT missing.  You are not authorized to make that request

}
```

Edit a Boat with PATCH

Allows you to edit from 1-3 of the attributes of a boat .

```
PATCH /boats/:boat_id
```

Request

Authorization:

Required. Valid JWT provided with user being in database, and also user being owner of boat provided

Request Parameters

Name	Type	Description	Required?
Boat_id	String	ID of the boat	Yes

Request Body

Required

The request body must contain at least one of the three attributes of a boat: name, type, length

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
Name	String	The name of the boat.	Yes
type	String	The type of the boat. E.g., Sailboat, Catamaran, etc.	Yes
length	Integer	Length of the boat in feet.	Yes

Request Body Example

```
{
  “name”: “Sea Witch”,
  “type”: “Catamaran”,
  “length”: 99
}
```

Response

Response Body Format

Success: JSON

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If the request body is missing, boat is not created and 400 status code returned If the input data provided is invalid, boat is not created and 400 status code returned

Failure	404 Not Found	No boat with this boat_id exists
Failure	403 Forbidden	JWT provided and user in database, but not owner of boat to be deleted
Failure	401 Unauthorized	JWT provided, user not in database Invalid JWT No JWT
Failure	406 Not Acceptable	If the request Accept header type is set and set to something other than JSON, boat is not edited and a 406 status code returned

Response Examples

- The `self` attribute will contain the live link to the REST resource corresponding to this boat. You must not store this attribute in Datastore.

Success

```
Status: 200 OK

{
  "id": "abc123",
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 99,
  "self": "https://<your-app>/boats/abc123"
}
```

Failure

```
Status: 400 Bad Request

{
"Error":  "The body of the request is missing"
}
```

```
Status: 404 Not Found

{
"Error":  "No boat with this boat_id exists"
}
```

Failure

```
Status: 403 Forbidden

{
“Error”:
“The boat you selected is owned by someone else.  You are not authorized to make that request”
}
```

Failure

```
Status: 401 Valid JWT but user not in database
```

```
{
“Error”:  {
Invalid user.  You are not in the database.  Please visit homepage to login
}
}
```

Failure

Status: 401 JWT invalid

```
{  “Error”:  Invalid JWT.  You are not authorized to make that request
}
```

Failure

Status: 401 JWT missing

```
{
“Error”:  JWT missing.  You are not authorized to make that request

}
```

Failure

Status: 406 Not Acceptable

```
{
“Error”:  “The requested MIMEtype is not allowed”
}
```

Edit a Boat with PUT

Allows you to edit from all of the attributes of a boat .

PUT /boats/:boat_id

Request

Authorization:

Required. Valid JWT provided with user being in database, and also user being owner of boat provided

Request Parameters

Name	Type	Description	Required?
Boat_id	String	ID of the boat	Yes

Request Body

Required

The request body must contain all three attributes of a boat: name, type, length

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
Name	String	The name of the boat.	Yes
type	String	The type of the boat. E.g., Sailboat, Catamaran, etc.	Yes
length	Integer	Length of the boat in feet.	Yes

Request Body Example

```
{
  “name”: “Sea Witch”,
  “type”: “Catamaran”,
  “length”: 99
}
```

Response

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

Failure	400 Bad Request	If the request body is missing, boat is not created and 400 status code returned If the input data provided is invalid, boat is not created and 400 status code returned
Failure	404 Not Found	No boat with this boat_id exists
Failure	403 Forbidden	JWT provided and user in database, but not owner of boat to be deleted
Failure	401 Unauthorized	JWT provided, user not in database Invalid JWT No JWT
Failure	406 Not Acceptable	If the request Accept header type is set and set to something other than JSON, boat is not edited and a 406 status code returned

Response Examples

Success

```
Status: 200
```

Failure

```
Status: 400 Bad Request

{
"Error":  "The body of the request is missing"
}
```

```
Status: 404 Not Found

{
"Error":  "No boat with this boat_id exists"
}
```

Failure

```
Status: 403 Forbidden

{
“Error”:
“The boat you selected is owned by someone else.  You are not authorized to make th
at request”
}
```

Failure

```
Status: 401 Valid JWT but user not in database

{
“Error”:  {
Invalid user.  You are not in the database.  Please visit homepage to login
}
}
```

Failure

```
Status: 401 JWT invalid
{  “Error”:  Invalid JWT.  You are not authorized to make that request
}
```

Failure

```
Status: 401 JWT missing

{
“Error”:  JWT missing.  You are not authorized to make that request

}
```

Create a Load

Allows you to create a new slip.

```
POST /loads
```

Request

Authentication requirement:

Required. A valid JWT with the user being a part of the database

Request Parameters

None

Request Body

Required

The body must have all three parts of a load

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
weight	Integer	The name of the boat.	Yes
content	string	What the load consists of	Yes
destination	string	Destination of load, can be city, state, or country	Yes

Request Body Example

```
{
  "weight": 10,
  "content": "Action figures",
  "destination": "London"
}
```

Response

Response Body Format

Success: JSON

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, the boat must not be created, and 400 status code must be returned.

Failure	401 Unauthorized	No JWT provided, 401 returned JWT valid, but user not in database, 401 returned JWT invalid, 401 returned
Failure	406 Not acceptable	If application/json not listed in Accept header

Response Examples

Success

```
Status: 201 Created

{
  “id”: “abc123”,
  “weight”: “Sea Witch”,
  “content”: “Action figures”,
   “carrier”: Null
  “destination”: “London”,
  “self”: “https://<your-app>/boats/abc123”
}
```

Failure

```
Status: 400 Bad Request

{
“Error”:  “The request object is missing at least one of the required attributes”
}
```

Failure

```
Status: 400 Bad Request

{
“Error”:  “The body of the request is missing”
}
```

Failure

```
Status: 401 Valid JWT but user not in database

{
“Error”:  {
Invalid user.  You are not in the database.  Please visit homepage to login
}
}
```

Failure

```
Status: 401 JWT missing

{
“Error”:  JWT missing.  You are not authorized to make that request

}
```

Failure

Status: 401 JWT invalid

```
{  “Error”:  Invalid JWT.  You are not authorized to make that request
}
```

Failure

Status: 406 Not Acceptable

```
{
“Error”:  “The requested MIMEtype is not allowed”
}
```

Get a load

Allows you to get an existing slip.

```
GET /loads/:load_id
```

Request

Authorization:

Not required

Request Parameters

Name	Type	Description	Required?

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No load with this load_id exists
Failure	406 Not Acceptable	Application/json not listed in Accept header

Response Examples

Success

```
Status: 200 OK

{
  "id": "123abc",
  "weight": 10,
   "carrier": null,
  “content”: “action figures”
  “destination”: “London”
  "self": "https://<your-app>/slips/123abc"
}
```

Failure

```
Status: 404 Not Found

{
"Error":  "No slip with this slip_id exists"
}
```

Failure

```
Status: 406 Not Acceptable

{
“Error”: “The requested MIMEtype is not allowed”
}
```

List all loads

List all the slips.

```
GET /loads
```

Request

Authorization:

Not required

Request Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	Application/json not listed in Accept header

Response Examples

Success

```
Status: 200 OK

[
{
  "id": "123abc",
  "weight": 10,
   "carrier": null,
  “content”: “action figures”
  “delivery_date”: 2019-12-20
  "self": "https://<your-app>/slips/123abc"
},
{
  "id": "456def",
  "weight": 100,
   "carrier": null,
  “content”: “gum”
  “destination”: “Pennsylvania”
  "self": "https://<your-app>/slips/456def"
}
]
```

Failure

```
Status: 406 Not Acceptable

{
“Error”: “The requested MIMEtype is not allowed”
}
```

Delete a load

Allows you to delete a slip. If the slip being deleted has a boat, the boat is now considered “at sea.”

```
DELETE /loads/:loads_id
```

Request

Authorization:

Required, valid JWT provided and user must be in the database

Request Parameters

None

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	JWT provided, user not in database Invalid JWT No JWT
Failure	404 Not Found	No boat with this boat_id exists

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 404 Not Found

{
"Error":  "No load with this load_id exists"
}
```

Failure

Status: 401 Valid JWT but user not in database

```
{
“Error”:  {
Invalid user.  You are not in the database.  Please visit homepage to login
}
}
```

Failure

Status: 401 JWT invalid

```
{  “Error”:  Invalid JWT.  You are not authorized to make that request
}
```

Failure

Status: 401 JWT missing

```
{
“Error”:  JWT missing.  You are not authorized to make that request

}
```

Edit a Load with PATCH

Allows you to edit from 1-3 of the attributes of a load .

```
PATCH /loads/:load_id
```

Request

Authorization:

Required. Valid JWT provided with user being in database

Request Parameters

Name	Type	Description	Required?

Request Body

Required

The request body must contain at least one of the three attributes of a boat: content, weight, destination

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
weight	Integer	The name of the boat.	Yes
content	string	What the load consists of	Yes
destination	string	Destination of load, can be city, state, or country	Yes

Request Body Example

```
{
  "weight": 10
}
```

Response

Response Body Format

Success: JSON

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If the request body is missing, load is not created and 400 status code returned
Failure	404 Not Found	No load with this load_id exists

Failure	401 Unauthorized	JWT provided, user not in database Invalid JWT No JWT
Failure	406 Not Acceptable	If the request Accept header type is set and set to something other than JSON, boat is not edited and a 406 status code returned

Response Examples

Success

```
Status: 200 OK

{
  “id”: “abc123”,
  “weight”: 10
  “content”: “Action figures”,
   “carrier”: Null
  “destination”: “London”,
  “self”: “https://<your-app>/boats/abc123”
}
```

Failure

```
Status: 400 Bad Request

{
"Error":  "The body of the request is missing"
}
```

```
Status: 404 Not Found

{
"Error":  "No load with this load_id exists"
}
```

Failure

```
Status: 401 Valid JWT but user not in database

{
“Error”:  {
Invalid user.  You are not in the database.  Please visit homepage to login
}
}
```

Failure

```
Status: 401 JWT invalid
{  “Error”:  Invalid JWT.  You are not authorized to make that request
}
```

Failure

```
Status: 401 JWT missing

{
“Error”:  JWT missing.  You are not authorized to make that request

}
```

Failure

```
Status: 406 Not Acceptable

{
“Error”:  “The requested MIMEtype is not allowed”
}
```

Edit a Load with PUT

Allows you to edit from all of the attributes of a boat .

```
PUT /loads/:load_id
```

Request

Authorization:

Required. Valid JWT provided with user being in database

Request Parameters

Name	Type	Description	Required?

Request Body

Required

The request body must contain all three attributes of a boat: weight, content, destination

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
weight	Integer	The name of the boat.	Yes
content	string	What the load consists of	Yes
destination	string	Destination of load, can be city, state, or country	Yes

Request Body Example

```
{
  "weight": 10,
  "content": "Action figures",
  "destination": "London"
}
```

Response

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If the request body is missing, load is not created and 400 status code returned If request body is missing any of the attributes of a load, load is not created and 400 status code returned

Failure	404 Not Found	No load with this load_id exists
Failure	401 Unauthorized	JWT provided, user not in database Invalid JWT No JWT

Response Examples

Success

Status: 200

Failure

Status: 400 Bad Request

```
{
"Error":  "The body of the request is missing"
}
```

Status: 404 Not Found

```
{
"Error":  "No load with this load_id exists"
}
```

Failure

Status: 403 Forbidden

```
{
“Error”:
“The boat you selected is owned by someone else.  You are not authorized to make th
at request”
}
```

Failure

Status: 401 Valid JWT but user not in database

```
{
“Error”:  {
Invalid user.  You are not in the database.  Please visit homepage to login
}
}
```

Failure

Status: 401 JWT invalid

```
{  “Error”:  Invalid JWT.  You are not authorized to make that request
}
```

Failure

```
Status: 401 JWT missing
{
“Error”:  JWT missing.  You are not authorized to make that request
}
```

Load assigned to boat

Load is added to boat.

```
PUT /boats/:boat_id/loads/:load_id
```

Request

Authorization:

Required. Valid JWT provided with user being in database, and also user being owner of boat provided

Request Parameters

None

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Succeeds only if a boat exists with this boat_id, a load exists with this load_id , the load is not already assigned, and user making request is owner of boat provided.
401	Unauthorized	JWT provided, user not in database Invalid JWT No JWT
Failure	403 Forbidden	A load with that load_id is already assigned to another boat. User is in the database, but not the owner of the boat
Failure	404 Not Found	No boat with this boat_id exists, and/or no load with this load_id exits.

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 403 Forbidden
{
"Error":  "A load with that load_id is already assigned to another boat"
}
```

```
Status: 404 Not Found
{
"Error":  "No boat/load with this boat_id/load_id exists"
}"
```

Failure

```
Status: 403 Forbidden
{
“Error”:
“The boat you selected is owned by someone else.  You are not authorized to make th
at request”
}
```

Failure

```
Status: 401 Valid JWT but user not in database
{
“Error”:  {
Invalid user.  You are not in the database.  Please visit homepage to login
}
```

Failure

```
Status: 401 JWT invalid
{  “Error”:  Invalid JWT.  You are not authorized to make that request
}
```

Failure

```
Status: 401 JWT missing
{
“Error”:  JWT missing.  You are not authorized to make that request
}
```

List all loads assigned to a specific boat

Boat has left the slip to go to sea. The slip is now empty.

```
GET /boats/:boat_id/loads
```

Request

Request Parameters

None

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	**Status Code**	**Notes**
Success	200 No Content	Succeeds only if a boat exists with this boat_id, a slip exists with this slip_id and this boat is at this slip.
Failure	404 Not Found	No boat with this boat_id is at the slip with this slip_id.
Failure	406 Not Acceptable	Application/json not in Accept header

Response Examples

Success

```
Status: 204 No Content

{
  “loads”:[
      "id": "456def",
       "weight": 100,
       "carrier": boat_id,
        “content”: “gum”
        “destination”: “New York”
        "self": "https://<your-app>/loads/456def"
]
}
```

Failure

```
Status: 404 Not Found

{
"Error":  "No boat with this boat_id exists"
}
```

Failure

```
Status: 406 Not Acceptable
```

```
{
“Error”: “The requested MIMEtype is not allowed”
}
```

Load removed from boat

Load is removed from boat.

```
DELETE /boats/:boat_id/loads/:load_id
```

Request

Authorization:

Required. Valid JWT provided with user being in database, and also user being owner of boat provided

Request Parameters

None

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	**Status Code**	**Notes**
Success	204 No Content	Succeeds only if a boat exists with this boat_id, a load exists with this load_id , the load is not already assigned, and user making request is owner of boat provided.
401	Unauthorized	JWT provided, user not in database Invalid JWT No JWT
Failure	403 Forbidden	A load with that load_id is already assigned to another boat. User is in the database, but not the owner of the boat
Failure	404 Not Found	No boat with this boat_id exists, and/or no load with this load_id exits.

Response Examples

Success

```
Status: 204 No Content
```

Failure

Status: 403 Forbidden

```
{
"Error":  "A load with that load_id is already assigned to another boat"
}
```

Status: 404 Not Found

```
{
"Error":  "No boat/load with this boat_id/load_id exists"
}
```

Failure

Status: 403 Forbidden

```
{
“Error”:
“The boat you selected is owned by someone else.  You are not authorized to make that request”
}
```

Failure

Status: 401 Valid JWT but user not in database

```
{
“Error”:  {
Invalid user.  You are not in the database.  Please visit homepage to login
}
}
```

Failure

Status: 401 JWT invalid

```
{  “Error”:  Invalid JWT.  You are not authorized to make that request
}
```

Failure

Status: 401 JWT missing

```
{
“Error”:  JWT missing.  You are not authorized to make that request
}
```