

Marina REST API Specifications

Spring 2022

Last Update: April 26. 11:15 am Pacific

Change log.....	2
Data Model	3
Description.....	4
Create a Boat (Protected)	6
Get a Boat (Protected)	8
List all Boats (Protected)	10
Delete a Boat (Protected)	13
Edit a Boat with PUT (Protected)	15
Edit a Boat with PATCH (Protected).....	17
Create a Load	19
Get a Load	21
List all Loads	22
Delete a Load	24
Edit a Load with PUT	25
Edit a Load with PATCH.....	27
Assign Load to a Boat (Protected).....	29
Load is Removed from Boat (Protected).....	31

Change log

Version	Change	Date
1.0	Initial version.	June 6, 2022

Important Note: All representations of resources have self-links. The self-attribute is not stored in the Datastore and is generated by the application/ API with the information provided in the request.

Data Model

The app stores three kinds of entities in Datastore: Boats, Loads and Users.

Boats

Property	Data Type	Required	Notes	Valid Values
id	Integer	Automatically generated upon creation.	The id of the boat. Datastore automatically generates it.	N/a
name	String	Yes	Name of the boat.	Battleship 2
type	String	Yes	Type of the boat.	Sailboat
length	Integer	Yes	The length of the boat in feet.	5
loads	Array of ints	No, an empty list will be generated upon creation.	This is an array of integers (id of loads).	N/a
owner	String	Automatically obtained from the authorization header.	This is the sub property from the jwt that was generated when the user logged in.	N/a

Loads

Property	Data Type	Required	Notes	Valid Values
id	Integer	Automatically generated upon creation.	The id of the load. Datastore automatically generates it.	N/a
volume	Integer	Yes	The volume of the load.	10
item	String	Yes	What items the load contains.	Lego Blocks
creation_date	String	Yes	The date the load was created. Was implemented as a string in this implementation.	10/18/2021
carrier	JSON object	No, an empty null object will be generated upon creation.	This json object refers to the boat that the load is currently on. It has the properties of id and name.	N/a

Users

Property	Data Type	Required	Notes	Valid Values
id	Integer	Automatically obtained upon first login.	The id of the load. Datastore automatically generates it.	N/a
first_name	String	Automatically generated upon first login.	First name of the user and this is taken from the user's email profile when they first log in.	N/a
last_name	String	Automatically	Last name of the user and this is taken from	N/a

		generated upon first login.	the user's email profile when they first log in.	
email	String	Automatically generated upon first login.	Email address of the user and this is taken from the user's email profile when they first log in.	N/a
unique_id	String	Automatically generated upon first login.	This is the sub property from the jwt that is generated when the user logs in.	N/a

Description

Relationship between non-user entities

The relationship between boats and loads is that a boat can carry multiple loads. However, a load can only be carried by one boat. Specifically, boats have a loads property which is a list of load ids to represent the loads on that boat. Conversely, a load has a carrier property which is a json object with the id and name of the boat that is carrying it. If the load has not been dropped on to a boat, it will return a null object for the carrier property. To set the relationship, you would send a PUT request to https://singmanb-hw2.uk.r.appspot.com/boats/boat_id/loads/load_id. Both both_id and load_id have to refer to entities that exist and the load can't be assigned to another boat. To remove the relationship, you would send a DELETE request to https://singmanb-hw2.uk.r.appspot.com/boats/boat_id/loads/load_id where load_id and boat_id refer to the ids of entities that exist. Lastly, a user can only modify the relationship for boats they own. It is a protected endpoint in that a user needs a valid authorization header (bearer token).

User entity

User accounts are handed through a 3rd party called Auth0. In this case, there is a URL on the homepage that an individual can use to sign in. For example, a user can sign in using their Gmail account. The user will be provided a corresponding JWT. If it is their first time logging in, an entry will be made in the Datastore as well. The unique identifier will be sub property in the corresponding jwt (under userinfo). The application will also use the information in the user's email profile to store a first name, last name, and email in the Datastore. All requests to a protected endpoint only require a bearer token and the value can be id_token found in the jwt. In this case, users own boats and all endpoints related to boats are protected. For protected resources, the application uses the sub property in the authorization header (jwt) and matches it with the owner property of boats. To map it to the user, the unique_id property of a user should match with the sub property of the authorization header(jwt).

Relationship between users and boats

The relationship between users and boats is that a user owns a boat. A user can own multiple boats, but a boat can only be owned by one user. Every endpoint related to boats is a protected resource. Specifically, the sub property of the authorization header has to match the owner property of the boats being affected or an error will be sent instead. This means that GET, POST /boats and GET, PUT, PATCH, DELETE /boats/boat_id need to have an authorization header (bearer token with value of

id_token from jwt). There is no relationship between users and loads. However, the relationship between boats and loads cannot be set unless the user owns that boat.

Create a Boat (Protected)

Allows you to create a new boat.

POST /boats

Request

Path Parameters

None

Request Body

Required

Request Headers

Authorization header required. It should be a bearer token with the value of id_token from the corresponding jwt.

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	The name of the boat.	Yes
type	The type of the boat. E.g., Sailboat, Catamaran, etc.	Yes
length	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 will not be created, and a 400 status code will be returned.
Failure	401 Bad Request	If the request does not have an authorization header, the boat will not be created and a 401-status code will be returned.
Failure	401 Bad Request	If the request does not have a valid authorization header, the boat will not be created and a 401-status code will be returned.

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. This value will be sent back in the response body as shown in the example.

Success

Status: 201 Created

```
{
  "id": 5698612453441536,
  "length": 28,
  "loads": [],
  "name": "Sea Witch",
  "owner": "google-oauth2|106460577505764504011",
  "self": "https://singmanb-hw2.uk.r.appspot.com/boats/5698612453441536",
  "type": "Catamaran"
}
```

Failure

Status: 400 Bad Request

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

Failure

Status: 401 Bad Request

```
{
  "code": "no auth header",
  "description": "Authorization header is missing"
}
```

Failure

Status: 401 Bad Request

```
{
  "code": "invalid_header",
  "description": "Invalid header. Use an RS256 signed JWT Access Token"
}
```

Get a Boat (Protected)

Allows you to get an existing boat

GET /boats/:boat_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Body

None

Request Headers

Authorization header required. It should be a bearer token with the value of id_token from the corresponding jwt.

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Bad Request	If the request does not have an authorization header, the boat will not be shown and a 401-status code will be returned.
Failure	401 Bad Request	If the request does not have a valid authorization header, the boat will not be shown and a 401-status code will be returned.
Failure	403 Forbidden	If the user does not own the boat they are trying to view, an error code and 403 status code will be returned.
Failure	404 Not Found	No boat with this boat_id exists

Response Examples

Success

Status: 200 OK

```
{
  "type": "Catamaran",
  "loads": [],
  "length": 28,
  "name": "Sea Witch",
  "owner": "google-oauth2|106460577505764504011",
  "id": 5698612453441536,
  "self": "https://singmanb-hw2.uk.r.appspot.com/boats/5698612453441536"
}
```


Failure

Status: 401 Bad Request

```
{  
  "code": "no auth header",  
  "description": "Authorization header is missing"  
}
```

Failure

Status: 401 Bad Request

```
{  
  "code": "invalid_header",  
  "description": "Invalid header. Use an RS256 signed JWT Access Token"  
}
```

Failure

Status: 403 Forbidden

```
{  
  "Error": "You do not have permission to view this boat as you are not the owner of this boat."  
}
```

Failure

Status: 404 Not Found

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

List all Boats (Protected)

List all the boats.

GET /boats

Request

Path Parameters

None

Request Body

None

Request Headers

Authorization header required. It should be a bearer token with the value of id_token from the corresponding jwt.

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Due to pagination, only 5 boats will be shown per page. There will be a next link for the next set of results and a total_items property for the total number of boats owned by this user. Only boats owned by the user will be shown.
Failure	401 Bad Request	If the request does not have an authorization header, the boat will not be shown and a 401-status code will be returned.
Failure	401 Bad Request	If the request does not have a valid authorization header, the boat will not be shown and a 401-status code will be returned.

Response Examples

Success

Status: 200 OK

```
{
  "boats":
  [{"loads": [],
    "owner": "google-oauth2|105847321833268803570",
    "length": 32,
    "type": "Large ship",
    "name": "The Champion",
    "id": 5135662500020224,
    "self": "https://singmanb-hw2.uk.r.appspot.com/boats/5135662500020224"}
  ],
}
```

```
{
  "owner": "google-oauth2|105847321833268803570",
  "length": 36,
  "type": "Large ship",
  "loads": [],
  "name": "Battle Ship 2",
  "id": 5632418417475584,
  "self": "https://singmanb-hw2.uk.r.appspot.com/boats/5632418417475584"
},
{
  "name": "Titan",
  "length": 60,
  "owner": "google-oauth2|105847321833268803570",
  "loads": [],
  "type": "Large Ship",
  "id": 5639966621171712,
  "self": "https://singmanb-hw2.uk.r.appspot.com/boats/5639966621171712"
},
{
  "owner": "google-oauth2|105847321833268803570",
  "length": 23,
  "type": "Medium ship",
  "loads": [],
  "name": "Noctis",
  "id": 5756975958720512,
  "self": "https://singmanb-hw2.uk.r.appspot.com/boats/5756975958720512"},
{
  "length": 10,
  "loads": [],
  "type": "Sail boat",
  "name": "River boat",
  "owner": "google-oauth2|105847321833268803570",
  "id": 5758387459457024,
  "self": "https://singmanb-hw2.uk.r.appspot.com/boats/5758387459457024"
}
],
"total_items": 6,
"next": "https://singmanb-hw2.uk.r.appspot.com/boats?limit=5&offset=5"
}
```

Failure

Status: 401 Bad Request

```
{
  "code": "no auth header",
  "description": "Authorization header is missing"
}
```

Failure

Status: 401 Bad Request

```
{  
  "code": "invalid_header",  
  "description": "Invalid header. Use an RS256 signed JWT Access Token"  
}
```

Delete a Boat (Protected)

Allows you to delete a boat. Note that if the boat currently has a load, deleting the boat will unload any loads that were loaded onto it.

DELETE /boats/:boat_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Body

None

Request Headers

Authorization header required. It should be a bearer token with the value of id_token from the corresponding jwt.

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Bad Request	If the request does not have an authorization header, the boat will not be deleted and a 401-status code will be returned.
Failure	401 Bad Request	If the request does not have a valid authorization header, the boat will not be deleted and a 401-status code will be returned.
Failure	403 Forbidden	If the user does not own the boat they are trying to delete, an error code and 403 status code will be returned.
Failure	404 Not Found	No boat with this boat_id exists

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Bad Request

```
{
  "code": "no auth header",
  "description": "Authorization header is missing"
}
```

```
}
```

Failure

Status: 401 Bad Request

```
{  
  "code": "invalid_header",  
  "description": "Invalid header. Use an RS256 signed JWT Access Token"  
}
```

Failure

Status: 403 Forbidden

```
{  
  "Error": "You do not have permission to delete this boat as you are not the owner of this boat."  
}
```

Failure

Status: 404 Not Found

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

Edit a Boat with PUT (Protected)

Allows you to edit a boat's properties. However, all three properties are required.

PUT /boats/boat_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Body

Required

Request Headers

Authorization header required. It should be a bearer token with the value of id_token from the corresponding jwt.

Request Body Format

JSON

Request JSON Attributes

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

Request Body Example

```
{
  "name": "Sea boat",
  "type": "Sailboat",
  "length": 12
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	303 See Other	Content-location is set to root URL where entity can be found. If the name is being changed, it has to be original and no other boat in the database should have this name.
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 will be returned.
Failure	400 Bad Request	If the request has more than the 3 required attributes, the boat will not be created, and a 400 status code will be returned.
Failure	401 Bad Request	If the request does not have an authorization header, the boat will not be created and a 401-status code will be

		returned.
Failure	401 Bad Request	If the request does not have a valid authorization header, the boat will not be created and a 401-status code will be returned.
Failure	403 Forbidden	If the user does not own the boat they are trying to change, an error code and 403 status code will be returned.
Failure	404 Not Found	No boat with this boat_id exists

Success

Status: 303 See Other

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 request object has too many attributes"
}
```

Failure

Status: 401 Bad Request

```
{
  "code": "no auth header",
  "description": "Authorization header is missing"
}
```

Failure

Status: 401 Bad Request

```
{
  "code": "invalid_header",
  "description": "Invalid header. Use an RS256 signed JWT Access Token"
}
```

Failure

Status: 403 Forbidden

```
{
  "Error": "You do not have permission to delete this boat as you are not the owner of this boat."
}
```

Failure

Status: 404 Not Found

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


Edit a Boat with PATCH (Protected)

Allows you to edit a boat's properties. Only one property is required.

PATCH /boats/boat_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Body

Required

Request Headers

Authorization header required. It should be a bearer token with the value of id_token from the corresponding jwt.

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	The name of the boat.	No. However, at least one property is required.
type	The type of the boat. E.g., Sailboat, Catamaran, etc.	No. However, at least one property is required.
length	Length of the boat in feet.	No. However, at least one property is required.

Request Body Example

```
{
  "name": "The MHS"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No content	If the name is being changed, it has to be original and no other boat in the database should have this name.
Failure	400 Bad Request	If the request does not have any of the correct attributes, the boat must not be created, and 400 status code will be returned.
Failure	400 Bad Request	If the request has more than 3 attributes, the boat will not be created, and a 400 status code will be returned.
Failure	401 Bad Request	If the request does not have an authorization header, the boat will not be created and a 401-status code will be

		returned.
Failure	401 Bad Request	If the request does not have a valid authorization header, the boat will not be created and a 401-status code will be returned.
Failure	403 Forbidden	If the user does not own the boat they are trying to change, an error code and 403 status code will be returned.
Failure	404 Not Found	No boat with this boat_id exists

Success

Status: 204 No Content

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 request object has too many attributes"
}
```

Failure

Status: 401 Bad Request

```
{
  "code": "no auth header",
  "description": "Authorization header is missing"
}
```

Failure

Status: 401 Bad Request

```
{
  "code": "invalid_header",
  "description": "Invalid header. Use an RS256 signed JWT Access Token"
}
```

Failure

Status: 403 Forbidden

```
{
  "Error": "You do not have permission to delete this boat as you are not the owner of this boat."
}
```

Failure

Status: 404 Not Found

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

Create a Load

Allows you to create a new load.

POST /loads

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
volume	The volume of the load.	Yes
item	What items the load contains	Yes
creation_date	The date the load was created	Yes

Request Body Example

```
{
  "volume": 5,
  "item": "LEGO Blocks",
  "creation_date": "10/18/2021"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is missing the attributes, the load will not be created, and 400 status code will be returned.

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. This value will be sent in the response body as shown in the example.
- The value of the attribute `carrier` is the ID of the boat that is carrying this load. If the load is unassigned, the value of `carrier` should be null.

Success

Status: 201 Created

```
{
  "carrier": null,
```

```
"creation_date": "10/18/2021",  
"id": 5639087159508992,  
"item": "LEGO Blocks",  
"self": "https://singmanb-hw2.uk.r.appspot.com/loads/5639087159508992",  
"volume": 5  
}
```

Failure

Status: 400 Bad Request

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

Get a Load

Allows you to get an existing load.

GET /loads/:load_id

Request

Path Parameters

Name	Description
load_id	ID of the load

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

Response Examples

Success

Status: 200 OK

```
{
  "carrier": null,
  "volume": 5,
  "item": "LEGO Blocks",
  "creation_date": "10/18/2021",
  "id": 5639087159508992,
  "self": "https://singmanb-hw2.uk.r.appspot.com/loads/5639087159508992"
}
```

Failure

Status: 404 Not Found

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

List all Loads

List all the loads.

GET /loads

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Due to pagination, only 5 loads will be shown per page. There will be a next link for the next set of results and a total_items property for the total number of loads.

Response Examples

Success

Status: 200 OK

```
{
  "loads":
  [
    {
      "creation_date": "10/19/2021",
      "item": "pans",
      "carrier": null,
      "volume": 20,
      "id": 5083500323536896,
      "self": "https://singmanb-hw2.uk.r.appspot.com/loads/5083500323536896"
    },
    {
      "creation_date": "10/19/2021",
      "volume": 20,
      "carrier": null,
      "item": "beds",
      "id": 5195437506035712,
      "self": "https://singmanb-hw2.uk.r.appspot.com/loads/5195437506035712"},
    {"creation_date": "10/18/2021",
      "item": "LEGO Blocks",
      "carrier": null,
      "volume": 5,
```

```
"id": 5639087159508992,
"self": "https://singmanb-hw2.uk.r.appspot.com/loads/5639087159508992"},
{"creation_date": "10/19/2021",
"item": "t-shirts",
"volume": 20,
"carrier": null,
"id": 5646450276958208,
"self": "https://singmanb-hw2.uk.r.appspot.com/loads/5646450276958208"},
{"carrier": null,
"volume": 15,
"item": "pants",
"creation_date": "10/18/2021",
"id": 5681414901071872,
"self": "https://singmanb-hw2.uk.r.appspot.com/loads/5681414901071872"}
],
"total_items": 6,
"next": "https://singmanb-hw2.uk.r.appspot.com/loads?limit=5&offset=5"
}
```

Delete a Load

Allows you to delete a load. If the load being deleted is currently loaded on a boat, the load is removed from that boat as well.

DELETE /loads/:load_id

Request

Path Parameters

Name	Description
load_id	ID of the load

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	404 Not Found	No load with this load_id exists

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found

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


Edit a Load with PUT

Allows you to edit a load's properties. However, all three properties are required.

PUT /loads/load_id

Request

Path Parameters

Name	Description
load_id	ID of the load

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
volume	The volume of the load.	Yes
item	What items the load contains	Yes
creation_date	The date the load was created	Yes

Request Body Example

```
{
  "volume": 45,
  "item": "Chairs",
  "creation_date": "03/18/2022"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	303 See Other	Content-location is set to root URL where entity can be found.
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 will be returned.
Failure	400 Bad Request	If the request has more than the 3 required attributes, the boat will not be created, and a 400 status code will be returned.
Failure	404 Not Found	No load with this load_id exists

Success

Status: 303 See Other

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 request object has too many attributes"  
}
```

Failure

Status: 404 Not Found

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

Edit a Load with PATCH

Allows you to edit a load's properties. Only one property is required.

PATCH /loads/load_id

Request

Path Parameters

Name	Description
load_id	ID of the load

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
volume	The volume of the load.	No. However, at least one property is required.
item	What items the load contains	No. However, at least one property is required.
creation_date	The date the load was created	No. However, at least one property is required.

Request Body Example

```
{
  "volume": 5
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No content	
Failure	400 Bad Request	If the request does not have any of the correct attributes, the boat must not be created, and 400 status code will be returned.
Failure	400 Bad Request	If the request has more than 3 attributes, the boat will not be created, and a 400 status code will be returned.
Failure	404 Not Found	No load with this load_id exists

Success

Status: 204 No Content

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 request object has too many attributes"
}
```

Failure

Status: 404 Not Found

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

Assign Load to a Boat (Protected)

A load is assigned to a boat.

PUT /boats/:boat_id/loads/:load_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat
load_id	ID of the load

Request Body

None

Request Headers

Authorization header required. It should be a bearer token with the value of id_token from the corresponding jwt.

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 and the load hasn't been assigned already as a load can only be assigned to one boat. A boat can have multiple loads. Additionally, the boat also has to be owned by the user requesting this change. The boat's load property is updated along with the load's carrier property.
Failure	401 Bad Request	If the request does not have an authorization header, the boat or load will not be changed and a 401-status code will be returned.
Failure	401 Bad Request	If the request does not have a valid authorization header, the boat or load will not be changed and a 401-status code will be returned.
Failure	403 Forbidden	If the user does not own the boat they are trying to change, an error code and 403 status code will be returned.
Failure	403 Forbidden	The load has already been assigned to a boat.
Failure	404 Not Found	The specified boat and/or load does not exist

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Bad Request

```
{
  "code": "no auth header",
  "description": "Authorization header is missing"
}
```

Failure

Status: 401 Bad Request

```
{
  "code": "invalid_header",
  "description": "Invalid header. Use an RS256 signed JWT Access Token"
}
```

Failure

Status: 403 Forbidden

```
{
  "Error": "You do not have permission to change this boat as you are not the owner of this boat."
}
```

Failure

Status: 403 Forbidden

```
{
  "Error": "The load is already on another boat"
}
```

Status: 404 Not Found

```
{
  "Error": "The specified boat and/or load does not exist"
}
```

Comments

- A load cannot be assigned to multiple boats.
- A boat can have multiple loads.
- If a load is already assigned to a boat, it must be removed first from that boat before it gets assigned to another boat.
- When a load has been assigned to a boat, the boat's load property gets updated. The load's carrier property also gets updated.
- Only the owner of the boat can add a load to their boat.

Load is Removed from Boat (Protected)

A load is removed from a boat.

DELETE /boats/:boat_id/loads/:load_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat
load_id	ID of the load

Request Headers

Authorization header required. It should be a bearer token with the value of id_token from the corresponding jwt.

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 and this load is loaded to this boat. Additionally, the user must be the owner of this boat. The boat's load property is updated along with the load's carrier property being updated to null .
Failure	401 Bad Request	If the request does not have an authorization header, the boat or load will not be changed and a 401-status code will be returned.
Failure	401 Bad Request	If the request does not have a valid authorization header, the boat or load will not be changed and a 401-status code will be returned.
Failure	403 Forbidden	If the user does not own the boat they are trying to change, an error code and 403 status code will be returned.
Failure	404 Not Found	No boat with this boat_id has a load with this load_id. This could be because no boat with this boat_id exists, or because no load with load_id exists, or even if both boat_id and load_id are valid, the boat with this boat_id does not have a load with this load_id

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Bad Request

```
{  
  "code": "no auth header",  
  "description": "Authorization header is missing"  
}
```

Failure

Status: 401 Bad Request

```
{  
  "code": "invalid_header",  
  "description": "Invalid header. Use an RS256 signed JWT Access Token"  
}
```

Failure

Status: 403 Forbidden

```
{  
  "Error": "You do not have permission to change this boat as you are not the owner of this boat."  
}
```

Failure

Status: 404 Not Found

```
{  
  "Error": " No boat with this boat_id is loaded with the load with this load_id"  
}
```


General Paths

If any request to an endpoint that will send back a response with a body does not have application/json in the Accept header, an error message will be sent back along with a 406 status code.

Failure

Status: 406 Not Acceptable

```
{  
  "Error": "This MIME type is not supported by this endpoint."  
}
```

For any request to an endpoint where the method is not supported such as a post request to /boats/boat_id, an error message will be returned along with a 405 status code.

Failure

Status: 405 Method Not Allowed

Method Not Allowed

The method is not allowed for the requested URL.