

# Portfolio Assignment: Final Project

URL: <http://cs-493-portfolio-tiua.uc.r.appspot.com>

CS 493: Cloud Application Development

Fall 2022

Oregon State University

Last Update: Dec 4. 11:15 pm Central

<a href="#">Change log</a>	2
<a href="#">Data Model</a>	2
<a href="#">View all Users</a>	4
<a href="#">Create a Boat</a>	5
<a href="#">View a Boat</a>	8
<a href="#">View all Boats</a>	10
<a href="#">Edit a Boat with PATCH</a>	12
<a href="#">Edit a Boat with PUT</a>	15
<a href="#">Delete a Boat</a>	18
<a href="#">Create a Load</a>	20
<a href="#">View a Load</a>	22
<a href="#">View all Loads</a>	24
<a href="#">Edit a Load with PATCH</a>	26
<a href="#">Edit a Load with PUT</a>	28

<a href="#">Delete a Load</a> .....	30
<a href="#">Assign Load to a Boat</a> .....	31
<a href="#">Remove a Load from a Boat</a> .....	33

## Change log

Version	Change	Date
1.0	Initial version.	December 3, 2022
2.0	Final version.	December 4, 2022

## Data Model

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

### Users

Property	Data Type	Notes
id	Integer	The id of the user automatically generated by datastore. Don't add it yourself as a property of the entity.
unique_id	String	The user's unique id which corresponds to the "sub" value in a user's JWT.
first_name	String	The user's first name according to Google's People API.
last_name	String	The user's last name according to Google's People API..

### Boats

Property	Data Type	Notes
id	Integer	The id of the boat automatically generated by datastore.
owner	String	The owner of the boat. This matches up with the "sub" value in a user's JWT and the unique_id of a user in the datastore.
name	String	Name of the boat.
type	String	Type of the boat. E.g., Sailboat, Catamaran, etc.
length	Integer	The length of the boat in feet.
loads	Array	A list of the ids of the load objects assigned to the boat.

### Loads

Property	Data Type	Notes
id	Integer	The id of the load. Datastore automatically generates it.
volume	Integer	The volume of the load.

carrier	Object	If there is a boat that is carrying this load, then an object with the boat's id and name. Otherwise null.
item	String	The item in the load. E.g., Legos, Video Games, etc.
creation_date	String	Date the load was created.

## View all Users (Unprotected Endpoint)

List all the users currently registered in the app.

GET /users

### Request

Path Parameters

None

Request Body

None

### Response

Response Body Format

JSON

### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

### Response Examples

#### Success

```
Status: 200 OK
[
  {
    "first_name": "Andrew",
    "last_name": "Tiu",
    "unique_id": "100789751072949765369",
    "id": "5637476211228672"
  },
  {
    "first_name": "James",
    "last_name": "Harden",
    "unique_id": "100789751072949765369",
    "id": "5637476211228672"
  }
]
```

## Create a Boat (Protected Endpoint)

Allows you to create a new boat. A valid JWT must be sent to make this request.

POST /boats

### Request

Path Parameters

None

Authorization Header

Bearer JWT

Request Body

Required

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	<p>If the request is missing any of the 3 required attributes, the boat must not be created, and 400 status code must be returned.</p> <p>You don't need to validate the values of the attributes and can assume that if the request contains any of the listed attributes, then that attribute's value is valid.</p> <p>You can also assume that the request will not contain any extraneous attribute (i.e., the request JSON will never contain any attribute that is not listed above).</p>

Failure	401 Unauthorized	If the request has a missing/invalid JWT, a 401 error will be returned.
Failure	403 Forbidden	If the request contains a name that is not unique, a 403 status code is returned.
Failure	406 Not Acceptable	If the client requests a content type that the server doesn't offer (i.e. not application/json), then a 406 status code is returned. Occurs when the client sets the accept header to a non application/json MIME type.
Failure	415 Unsupported Media Type	If the request sent is not a supported MIME type (i.e. not in application/json format), a 415 status code is returned. Occurs when the client sets the content-type header in the request to a non application/json MIME type.

### Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. The owner value will be populated from the "sub" attribute of a valid JWT.
- Loads attribute is an array of the current loads on the ship. When a boat is created this attribute is initialized as an empty array.
- Self attribute contains a link that goes to the direct location to find the resource corresponding to the newly created boat. This link is generated on the fly and is not stored on the datastore.

### Success

#### Status: 201 Created

```
{
  "id": 123,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28
  "loads": []
  "self": "https://<your-app>/boats/123"
}
```

### Failure

#### Status: 400 Bad Request

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

### Failure

#### Status: 401 Unauthorized

```
{
  "Error": "JWT is invalid / missing."
}
```

*Failure*

**Status: 403 Forbidden**

```
{  
  "Error": "The name given is not unique and already exists."  
}
```

*Failure*

**Status: 406 Not Acceptable**

```
{  
  "Error": "The requested response content type is not supported."  
}
```

*Failure*

**Status: 415 Unsupported Media Type**

```
{  
  'Error': 'Server only accepts application/json data.'  
}
```

## View a Boat (Protected Endpoint)

Allows you to view an existing boat that you are the owner of.

GET /boats/:boat\_id

### Request

#### Path Parameters

Name	Description
boat_id	ID of the boat

### Authorization Header

Bearer JWT

### Request Body

None

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	If the request has a missing/invalid JWT, a 401 error will be returned.
Failure	403 Forbidden	Access to this boat is forbidden because the requester does not own the specified boat.
Failure	404 Not Found	No boat with this boat_id exists.
Failure	406 Not Acceptable	If the client requests a content type that the server doesn't offer (i.e. not application/json), then a 406 status code is returned. Occurs when the client sets the accept header to a non application/json MIME type.

### Response Examples

#### Success

**Status: 200 OK**

```
{
  "id": 123,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28
  "loads": []
  "self": "https://<your-app>/boats/123"
}
```



*Failure*

**Status: 401 Unauthorized**

```
{  
  "Error": "JWT is invalid / missing."  
}
```

*Failure*

**Status: 403**

```
{  
  "Error": "You are not authorized to view this boat."  
}
```

*Failure*

**Status: 404 Not Found**

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

*Failure*

**Status: 406 Not Acceptable**

```
{  
  "Error": "The requested response content type is not supported."  
}
```

## View all Boats (Protected Endpoint)

List all the boats belonging to the owner specified in the JWT.

GET /boats
------------

### Request

Path Parameters

None

Authorization Header

Bearer JWT

Request Body

None

### Response

Response Body Format

JSON

### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	If the request has a missing/invalid JWT, a 401 error will be returned.

### Response Examples

- Pagination is implemented, so 5 boats are returned per page, and if there are more than 5 boats total then a “next” link should be in the response that will take the user to the next 5 boats.

#### Success

**Status: 200 OK**

```
[
  {
    "id": 123,
    "name": "Sea Witch",
    "type": "Catamaran",
    "length": 28,
    "owner": 28233445
    "loads": [],
    "self": "https://<your-app>/boats/123"
  },
  {
    "id": 456,
    "name": "Adventure",
    "type": "Lifeboat",
    "length": 50,
```

```
"owner": 28233445
"loads": [123, 234],
"self": "https://<your-app>/boats/456"
},
{
  "id": 789,
  "name": "Hocus Pocus",
  "type": "Sailboat",
  "length": 100,
  "owner": 28233445
  "loads": [],
  "self": "https://<your-app>/boats/789"
}
{
  "id": 223,
  "name": "Big Chungus",
  "type": "Cruise",
  "length": 100,
  "owner": 28233445
  "loads": [],
  "self": "https://<your-app>/boats/789"
}
{
  "id": 229,
  "name": "Fasty",
  "type": "Raceboat",
  "length": 23,
  "owner": 28233445
  "loads": [],
  "self": "https://<your-app>/boats/789"
}
]
"next": "https://<your-app>/boats?cursor=...."
```

#### Failure

##### Status: 401 Unauthorized

```
{
  "Error": "JWT is invalid / missing."
}
```

## Edit a Boat with PATCH (Protected Endpoint)

Allows you to edit a subset of a boat's attributes while the other attributes remain unchanged.

PATCH /boats/:boat_id
-----------------------

### Request

#### Path Parameters

Name	Data Type	Description	Required?
boat_id	String	ID of the boat	Yes

#### Authorization Header

Bearer JWT

#### Request Body

Required

#### Request Body Format

application/json

#### Request JSON Attributes

Name	Data Type	Description	Required?
name	String	The name of the boat. Name must be unique.	No if at least one other attribute is provided.
type	String	The type of the boat. E.g., Sailboat, Catamaran, etc.	No if at least one other attribute is provided.
length	Integer	Length of the boat in feet.	No if at least one other attribute is provided.

#### Request Body Example

<pre>{   "type": "Fishing", }</pre>
-------------------------------------

### Response

#### Response Body Format

application/json

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If the request is invalid, a 400 status code is returned. A request is considered invalid if all attributes are missing.
Failure	401 Unauthorized	If the request has a missing/invalid JWT, a 401 error will be returned.
Failure	403 Forbidden	If the request contains a name that is not unique, or the owner of the JWT does not own the boat to be edited, a 403 status code is returned.

Failure	404 Not Found	No boat with this boat_id exists
Failure	406 Not Acceptable	If the client requests a content type that the server doesn't offer (i.e. not application/jsonl), then a 406 status code is returned. Occurs when the client sets the accept header to a non application/json MIME type.
Failure	415 Unsupported Media Type	If the request sent is not a supported MIME type (i.e. not in application/json format), a 415 status code is returned. Occurs when the client sets the content-type header in the request to a non application/json MIME type.

## Response Examples

### Success

#### Status: 200 OK

```
{
  "id": 123,
  "name": "Sea Witch",
  "type": "Fishing",
  "length": 28
  "self": "https://<your-app>/boats/123"
```

### Failure

#### Status: 400 Bad Request

```
{
  "Error": "The request object is missing required attributes."
}
```

### Failure

#### Status: 401 Unauthorized

```
{
  "Error": "JWT is invalid / missing."
}
```

### Failure

#### Status: 403 Forbidden

```
{
  "Error": "You are not authorized to edit this boat."
}
```

### Failure

#### Status: 403 Forbidden

```
{
  "Error": "The name given is not unique and already exists."
}
```

### Failure

#### Status: 404 Not Found

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

*Failure*

**Status: 406 Not Acceptable**

```
{  
  "Error": "The requested response content type is not supported."  
}
```

*Failure*

**Status: 415 Unsupported Media Type**

```
{  
  "Error": "Server only accepts application/json data."  
}
```

## Edit a Boat with PUT (Protected Endpoint)

Allows you to edit all the attributes of a boat that you own.

PUT /boats/:boat\_id

### Request

#### Path Parameters

Name	Data Type	Description	Required?
boat_id	String	ID of the boat	Yes

#### Authorization Header

Bearer JWT

### Request Body

Required

#### Request Body Format

application/json

#### Request JSON Attributes

Name	Data Type	Description	Required?
name	String	The name of the boat. Must be unique.	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": "Big Chungus",
  "type": "Trawler",
  "length": "500",
}
```

### Response

#### Response Body Format

application/json

#### Response Statuses

Outcome	Status Code	Notes
Success	303 See Other	This is not the standard status code that is usually returned, but it is necessary in this case. Location header in the response contains the URL of the updated boat.
Failure	400 Bad Request	If the request is invalid, a 400 status code is returned. A request is considered invalid if any of the attributes are missing.
Failure	401 Unauthorized	If the request has a missing/invalid JWT, a 401 error will be returned.

Failure	403 Forbidden	If the request contains a name that is not unique, or the boat does not belong to the owner of the JWT, a 403 status code is returned.
Failure	404 Not Found	No boat with this boat_id exists.
Failure	406 Not Acceptable	If the client requests a content type that the server doesn't offer (i.e. not application/json), then a 406 status code is returned. Occurs when the client sets the accept header to a non application/json MIME type.
Failure	415 Unsupported Media Type	If the request sent is not a supported MIME type (i.e. not in application/json format), a 415 status code is returned. Occurs when the client sets the content-type header in the request to a non application/json MIME type.

## Response Examples

### Success

#### Status: 303 OK

Note: URL of updated boat should be in the location header and should be "https://<your-app>/boats/123"

```
{
  "id": 123,
  "name": "Big Chungus",
  "type": "Trawler",
  "length": 500
  "self": "https://<your-app>/boats/123"
}
```

### Failure

#### Status: 400 Bad Request

```
{
  "Error": "The request object is missing required attributes."
}
```

### Failure

#### Status: 401 Unauthorized

```
{
  "Error": "JWT is invalid / missing."
}
```

### Failure

#### Status: 403 Forbidden

```
{
  "Error": "You are not authorized to edit this boat."
}
```

### Failure

#### Status: 403 Forbidden



```
{  
  "Error": "The name given is not unique and already exists."  
}
```

*Failure*

**Status: 404 Not Found**

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

*Failure*

**Status: 406 Not Acceptable**

```
{  
  "Error": "The requested response content type is not supported."  
}
```

*Failure*

**Status: 415 Unsupported Media Type**

```
{  
  "Error": "Server only accepts application/json data."  
}
```

## Delete a Boat (Protected Endpoint)

Allows you to delete a boat that you own. Note that if the boat has any loads on it, then the loads on the ship will be unloaded and become unassigned in the datastore.

DELETE /boats/:boat_id
------------------------

### Request

#### Path Parameters

Name	Description
boat_id	ID of the boat

#### Authorization Header

Bearer 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	
Failure	401 Unauthorized	If the request has a missing/invalid JWT, a 401 error will be returned.
Failure	403 Forbidden	If the owner of the boat does not match with the owner of the JWT, deletion of the boat is not allowed.
Failure	404 Not Found	No boat with this boat_id exists

#### Response Examples

##### Success

Status: 204 No Content
------------------------

##### Failure

###### Status: 401 Unauthorized

```
{
  "Error": "JWT is invalid / missing."
}
```

##### Failure

###### Status: 403 Forbidden

```
{
  "Error": "You are not authorized to delete this boat."
}
```

```
}
```

#### *Failure*

**Status: 404 Not Found**

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

## Create a load (Unprotected Endpoint)

Allows you to create a new load.

POST /loads
-------------

### Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
item	String	The item that makes up the load.	Yes
volume	Integer	The volume of the load.	Yes
creation_date	String	The date the load was created.	Yes

Request Body Example

```
{
  "item": "LEGO Blocks"
  "volume": 6
  "creation_date": "10/10/2022"
}
```

### 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 volume, item or creation date attributes, then an error is returned.  Validation of the attributes is not required and it can be assumed that if they are specified then they are valid.  It is assumed that the request will not contain any extraneous attribute (i.e., the request JSON will never contain any attributes other than volume, item or load).
Failure	406 Not Acceptable	If the client requests a content type that the server doesn't offer (i.e. not application/json), then a 406 status code is returned. Occurs when the client sets the accept header to a non application/json MIME type.

Failure	415 Unsupported Media Type	If the request sent is not a supported MIME type (i.e. not in application/json format), a 415 status code is returned. Occurs when the client sets the content-type header in the request to a non application/json MIME type.
---------	----------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. This value is sent in the response body as shown in the example.
- The value of the attribute `carrier` is an object containing the ID and name of the boat carrying the item. If there is currently no boat carrying this load then the carrier is null. When a load is created, carrier is initialized as null.
- Self attribute contains a link that goes to the direct location to find the resource corresponding to the newly created load. This link is generated on the fly and is not stored on the datastore.

### Success

#### Status: 201 Created

```
{
  "id": 6,
  "item": "LEGO Blocks",
  "volume": 5,
  "creation_date": "10/10/2021",
  "carrier": null,
  "self": "https://<your-app>/loads/6"
}
```

### Failure

#### Status: 400 Bad Request

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

### Failure

#### Status: 406 Not Acceptable

```
{
  "Error": "The requested response content type is not supported."
}
```

### Failure

#### Status: 415 Unsupported Media Type

```
{
  "Error": "Server only accepts application/json data."
}
```

## View a Load (Unprotected Endpoint)

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
Failure	406 Not Acceptable	If the client requests a content type that the server doesn't offer (i.e. not application/json), then a 406 status code is returned. Occurs when the client sets the accept header to a non application/json MIME type.

#### Response Examples

##### Success

###### Status: 200 OK

```
{
  "id": 6,
  "item": "LEGO Blocks",
  "volume": 5,
  "creation_date": "10/10/2021"
  "carrier": null,
  "self": "https://<your-app>/loads/6"
}
```

##### Failure

###### Status: 404 Not Found

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

##### Failure

###### Status: 406 Not Acceptable

```
{
```

```
"Error": "The requested response content type is not supported."  
}
```

## View all Loads (Unprotected Endpoint)

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	

### Response Examples

- Pagination is implemented, so 5 loads are returned per page, and if there are more than 5 loads total then a “next” link should be in the response that will take the user to the next 5 loads.

#### Success

**Status: 200 OK**

```
[
{
  "id": 6,
  "item": "LEGO Blocks",
  "volume": 5,
  "creation_date": "10/10/2021",
  "carrier": null,
  "self": "https://<your-app>/loads/6"
},
{
  "id": 7,
  "item": "Pokemon Cards",
  "volume": 6,
  "creation_date": "11/10/2021",
  "carrier": 500,
  "self": "https://<your-app>/loads/7"
}
{
  "id": 8,
  "item": "Basketball"
  "volume": 500
  "creation_date": "12/10/2021"
```



```
"carrier": null
"self": "https://<your-app>/loads/8"
}
{
  "id": 9,
  "item": "Apple iPhone"
  "volume": 200
  "creation_date": "12/22/2021"
  "carrier": null
  "self": "https://<your-app>/loads/9"
}
{
  "id": 10,
  "item": "PS5"
  "volume": 15
  "creation_date": "03/10/2021"
  "carrier": null
  "self": "https://<your-app>/loads/10"
}
]
"next": "https://<your-app>/loads?cursor=....."
```

## Edit a Load with PATCH (Unprotected Endpoint)

Allows you to edit a subset of a load's attributes while the other attributes remain unchanged.

PATCH /loads/:load\_id

### Request

#### Path Parameters

Name	Data Type	Description	Required?
load_id	String	ID of the load	Yes

### Request Body

Required

#### Request Body Format

application/json

#### Request JSON Attributes

Name	Data Type	Description	Required?
item	String	The item in the load.	No if at least one other attribute is provided.
volume	Integer	The volume of the load.	No if at least one other attribute is provided.
creation_date	String	The creation date of the load.	No if at least one other attribute is provided.

#### Request Body Example

```
{
  "item": "Xbox One",
}
```

### Response

#### Response Body Format

application/json

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If the request is invalid, a 400 status code is returned. A request is considered invalid if all attributes are missing.
Failure	404 Not Found	No load with this load_id exists
Failure	406 Not Acceptable	If the client requests a content type that the server doesn't offer (i.e. not application/json), then a 406 status code is returned. Occurs when the client sets the accept header to a non application/json MIME type.
Failure	415 Unsupported Media Type	If the request sent is not a supported MIME type (i.e. not in application/json format), a 415 status code is returned.

		Occurs when the client sets the content-type header in the request to a non application/json MIME type.
--	--	---------------------------------------------------------------------------------------------------------

## Response Examples

### Success

#### Status: 200 OK

```
{
  "id": 123,
  "item": "Xbox One",
  "volume": 15,
  "creation_date": "03/30/2022",
  "carrier": null,
  "self": "https://<your-app>/boats/123"
}
```

### Failure

#### Status: 400 Bad Request

```
{
  "Error": "The request object is missing required attributes."
}
```

### Failure

#### Status: 404 Not Found

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

### Failure

#### Status: 406 Not Acceptable

```
{
  "Error": "The requested response content type is not supported."
}
```

### Failure

#### Status: 415 Unsupported Media Type

```
{
  "Error": "Server only accepts application/json data."
}
```

## Edit a Load with PUT (Unprotected Endpoint)

Allows you to edit all the attributes of a load.

PUT /loads/:load\_id

### Request

#### Path Parameters

Name	Data Type	Description	Required?
load_id	String	ID of the load.	Yes

### Request Body

Required

#### Request Body Format

application/json

#### Request JSON Attributes

Name	Data Type	Description	Required?
item	String	The item in the load.	Yes
volume	Integer	The volume of the load.	Yes
creation_date	String	The creation date of the load.	Yes

### Request Body Example

```
{
  "item": "Roomba",
  "volume": 50,
  "creation_date": "08/25/2021",
}
```

### Response

#### Response Body Format

application/json

#### Response Statuses

Outcome	Status Code	Notes
Success	303 See Other	This is not the standard status code that is usually returned, but it is necessary in this case. Location header in the response contains the URL of the updated load.
Failure	400 Bad Request	If the request is invalid, a 400 status code is returned. A request is considered invalid if any of the attributes are missing.
Failure	404 Not Found	No load with this load_id exists.
Failure	406 Not Acceptable	If the client requests a content type that the server doesn't offer (i.e. not application/json), then a 406 status

		code is returned. Occurs when the client sets the accept header to a non application/json MIME type.
Failure	415 Unsupported Media Type	If the request sent is not a supported MIME type (i.e. not in application/json format), a 415 status code is returned. Occurs when the client sets the content-type header in the request to a non application/json MIME type.

## Response Examples

### Success

#### Status: 303 OK

Note: URL of updated load should be in the location header and should be

"https://<your-app>/loads/123"

```
{
  "id": 123,
  "item": "Fitbit",
  "volume": 50,
  "creation_date": "08/15/2022",
  "carrier": null
  "self": "https://<your-app>/loads/123"
}
```

### Failure

#### Status: 400 Bad Request

```
{
  "Error": "The request object is missing required attributes."
}
```

### Failure

#### Status: 404 Not Found

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

### Failure

#### Status: 406 Not Acceptable

```
{
  "Error": "The requested response content type is not supported."
}
```

### Failure

#### Status: 415 Unsupported Media Type

```
{
  "Error": "Server only accepts application/json data."
}
```

## Delete a Load (Unprotected Endpoint)

Allows you to delete a load. If the load being deleted has a carrier, the load will be deleted from the list of loads on the boat.

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  <pre>{   "Error": "No load with this load_id exists" }</pre>
---------------------------------------------------------------------------------------------------

## Assign Load to a Boat (Protected Endpoint)

Load has a load assigned to it.

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

### 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 is initially unassigned.
Failure	401 Unauthorized	The JWT is invalid/missing.
Failure	403 Forbidden	The load is already loaded on another boat or the owner of the boat does not match the owner of the JWT.
Failure	404 Not Found	The specified boat and/or load does not exist.

#### Response Examples

##### Success

<b>Status: 204 No Content</b>
-------------------------------

##### Failure

<b>Status: 401 Unauthorized</b>
<pre>{   "Error": "JWT is invalid / missing." }</pre>

##### Failure

<b>Status: 403 Forbidden</b>
<pre>{   "Error": "You are not authorized to perform this action." }</pre>

### *Failure*

**Status: 403 Forbidden**

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

**Status: 404 Not Found**

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

### *Comment*

- A load cannot be assigned to multiple boats at the same time. It can only be on one boat at once.



## Remove a Load from a Boat (Protected Endpoint)

Load is removed from the boat. The boat no longer has that load.

DELETE /boats/:boat_id/loads/:load_id
---------------------------------------

### Request

#### Path Parameters

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

#### Authorization Header

Bearer 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 initially unassigned.
Failure	401 Unauthorized	The JWT is invalid/missing.
Failure	403 Forbidden	The owner of the JWT does not match with the owner of the boat.
Failure	404 Not Found	No boat with this load_id is at the boat with this boat_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, but the load with this load_id is not on the boat with this boat_id and is instead on another boat.

#### Response Examples

##### Success

Status: 204 No Content
------------------------

*Failure*

**Status: 401 Unauthorized**

```
{  
"Error": "JWT is invalid / missing."  
}
```

*Failure*

**Status: 403 Forbidden**

```
{  
"Error": "You are not authorized to perform this action."  
}
```

*Failure*

**Status: 404 Not Found**

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