

# Shipping Port

Author: Christian McVerry

Last Updated: August 31, 2022

URL: <https://shipping-port-6543.ue.r.appspot.com/>

## Table of Contents

<a href="#"><u>Change Log</u></a>	3
<a href="#"><u>Purpose</u></a>	4
<a href="#"><u>Data Model</u></a>	5
<a href="#"><u>Data Properties</u></a>	6
<a href="#"><u>Data Constraints</u></a>	6
<a href="#"><u>Non-User Entities Relationship</u></a>	6
<a href="#"><u>User Entity</u></a>	6
<a href="#"><u>View Users</u></a>	7
<a href="#"><u>Login User</u></a>	9
<a href="#"><u>Create a Boat</u></a>	12
<a href="#"><u>View User's Boats</u></a>	15
<a href="#"><u>View a Boat</u></a>	18
<a href="#"><u>Delete a Boat</u></a>	20
<a href="#"><u>Edit a Boat (PUT)</u></a>	22
<a href="#"><u>Edit a Boat (PATCH)</u></a>	25
<a href="#"><u>Create a Load</u></a>	28
<a href="#"><u>View Loads</u></a>	30
<a href="#"><u>View a Load</u></a>	32
<a href="#"><u>Delete a Load</u></a>	34
<a href="#"><u>Edit a Load (PUT)</u></a>	35
<a href="#"><u>Edit a Load (PATCH)</u></a>	38
<a href="#"><u>Assign a Load to a Boat</u></a>	41
<a href="#"><u>Remove a Load from a Boat</u></a>	43

## Change Log

Version	Change	Date
1.0	Initial version.	June 5, 2022
1.1	Updated/edited some sections and added the Purpose section.	June 28, 2022
1.2	Updated out-of-date examples, minor edits.	August 22, 2022
1.3	Fixed issue with table of contents links, minor edits.	August 31, 2022

## Purpose

Shipping Port is a functional deployed implementation of a RESTful API. This API allows users to create, modify, delete, and manipulate two related data entities boats and loads that are stored on Google Datastore. Interaction with the API occurs via HTTP requests made to the API's various endpoints. A user who has authenticated with the API can freely create their own boats, loads, and perform a couple interactions between the two related data entities.

To authenticate with the API, users must visit <https://shipping-port-6543.ue.r.appspot.com/> and click the login link. Upon clicking login, a user is redirected to an Auth0 authentication page. The API's authentication is handled by Auth0, a 3<sup>rd</sup> party authentication/authorization service. A user can either choose to authenticate using an Auth0 or Google account.

**Note:** By authenticating with a Google account, a user shares their name, email address, language preference, and profile picture with Auth0.

Upon authenticating, a user is redirected to an info page that lists the user's name, a unique ID that the API uses to identify the user, and a JSON Web Token (JWT). Now a user is ready to start interacting with the API. All HTTP requests that interact with boat entities require an Authorization header containing "Bearer {user's JWT}". Examples of HTTP requests can be found in provided Postman JSON files. JWTs do have expiration periods, after which they become invalid. A user must visit <https://shipping-port-6543.ue.r.appspot.com/> whenever they need a new valid JWT. Alternatively, a user who has authenticated using an Auth0 account can retrieve a new valid JWT via a login request to the API with their login credentials. Users who have authenticated using a Google account will need to revisit the URL and authenticate to receive a new valid JWT.

## Data Model

The application stores three kinds of entities in Google Datastore: boats, loads, and users.

### Boats

Property	Data Type	Notes
id	Integer	The id of the boat. Automatically generated by 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 of Objects	Contains id values of carried loads.
user	String	Contains user_id associated with the user who created the boat entity.

### Loads

Property	Data Type	Notes
id	Integer	The id of the load. Automatically generated by Datastore.
name	String	Name of the load.
quantity	String	Quantity of the load.
weight	Integer	Weight of the load.
carrier	Object	Contains id of boat carrying load.

### Users

Property	Data Type	Notes
id	Integer	The id of the user entity. Automatically generated by Datastore.
name	String	Name of user, generated as <i>name</i> in the user's JWT
user_id	String	Unique value generated as <i>sub</i> in the user's JWT that is used as the primary means of identifying a user
boats	Array of Objects	Contains id values of boats tied to the user

## Data Properties

*name*, *type*, and *length* are required properties of a boat entity that a user must include within a JSON-formatted request body upon creating or editing a boat entity. *id*, *loads*, and *user* are handled by Google Datastore and the API. Similarly, *name*, *quantity*, and *weight* are required properties of a load entity that a user must include with a JSON-formatted request body upon creating or editing a load entity. *id* and *carrier* are handled by the Google datastore and API. A user entity with properties *id*, *name*, *user\_id*, and *boats* are automatically created upon a user authenticating with the app.

## Data Constraints

There are not many data constraints enforced by the API. The API works under the assumption that request bodies contain attributes whose data types are consistent with the API's data model. I.e., there is no verification that checks that the value passed with *id* property for an entity is an integer.

## Non-User Entities Relationship

The two non-user entities in this API are boats and loads. Boats are a protected resource, for which reason to interact with or create a boat a user must include a valid JWT in the Authorization header of all requests that interact with boats. Loads are not a protected resource. A user does not need a valid JWT to interact with loads only.

Boats and loads have a many-to-one relationship. A boat can carry multiple loads, but a load can only be carried by one boat. Endpoints methods for boat and load relationship interactions (assign a load to a boat or remove a load from a boat) require an Authorization header with a valid JWT, as a protected resource, a boat, is being accessed. I.e., only the user who owns the boat can load or remove a load from a specified owned boat. In the relationship between a boat and load are represented through a boat's *loads* property and a load's *carrier* property.

## User Entity

A user entity is automatically generated and associated with a user upon authenticating with the application. The application uses Auth0 for handling user authentication. User JWTs are generated by Auth0 and returned by the API. The *user\_id* property is used as the unique identifier in the user entity. The *user\_id* property directly corresponds to the *sub* value within the JWT corresponding to a user. Upon initial user authentication, the JWT's *sub* value is copied and assigned to the *user\_id* property of a new user entity. While a JWT's *id\_token* value will vary between subsequent user authentications, *sub* will remain the constant.

The relationship between users and boats is a many-to-one relationship. A user can be associated with multiple boats, but a boat can only be associated with one user. The *user* property in a boat entity and the *boats* property in a user entity define the relationship between these two entities.

A valid JWT is required in the Authorization header for requests that interact with boats. The API verifies that a supplied JWT is valid and compares the JWT's *sub* value to a boat's *user* property. An interaction

with a boat entity can only be successful if the supplied JWT is both valid and its *sub* value is the same as the boat's *user* property value.

## View Users

Retrieves all users who have authenticated with the app

Endpoint Protected: No

GET /users
------------

### Request

Path Parameters

None

Request Body

None

Request Headers

Required:

Accept: application/json

### Response

MIME Types for Response Body

Success: application/json

Failure: application/json

### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	If the request's 'Accept' header is not MIME type 'application/json'

### Response Examples

#### Success

Status: 200 OK
<pre>{   "users": [     {       "boats": [],       "name": "seconduser@foo.com",       "user_id": "auth0 629ceeae3ec8fd0069c95923",       "collection_count": "1 of 2"     },     {       "boats": [</pre>

```
        "id": 6309930852155392,  
  
    self": "https://shipping-port-6543.ue.r.appspot.com/boats/630993085215539"  
  
        }  
    ],  
    "name": "firstuser@foo.com",  
    "user_id": "auth0|629a853be9b4960069f37841",  
    "collection_count": "2 of 2"  
  }  
]  
}
```

#### *Failure*

Status: 406 Not Acceptable

```
{  
  "Error": "Only JSON data can be returned in response object"  
}
```



## Login User

Retrieves JWT from Auth0 for a user based on the request body properties username and password.

Note: the user must have authenticated with the application via the web app login (visit <https://portfolio-api-mcverryc.ue.r.appspot.com/> ) before using this request to retrieve a JWT. *id\_token* returned in the response body is the user's JWT. It is not required that a user uses this API request to retrieve a JWT. A JWT corresponding to the user is also displayed when a user performs a web app login.

Endpoint Protected: No

POST /login
-------------

### Request

#### Path Parameters

None

#### Request Body

Required

#### Request Headers

Required:

Content-Type: application/json

Accept: application/json

#### MIME Types for Request Body

application/json

#### Request JSON Attributes

Name	Description	Required?
username		Yes
password		Yes

#### Request Body Example

```
{
  "username": "firstuser@foo.com",
  "password": "43it#Hie"
}
```

### Response

#### MIME Types for Response Body

Success: application/json

Failure: application/json

## Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, or any of the required attributes have invalid values.
Failure	415 Unsupported Media Type	If the request's 'Content-Type' header is not MIME type 'application/json'
Failure	406 Not Acceptable	If the request's 'Accept' header is not MIME type 'application/json'

## Response Examples

- "id\_token" in response body is the JWT.

### Success

<b>Status: 201 Created</b>
<pre>{   "access_token": "eyJhbGciOiJkaXIiLCJlbmMiOiJBMjU2R0NNIiwiaXNzIjoiaHR0cHM6Ly9wb3J0Zm9saW8tYXBpLnVzLmFldGgwLmNvbS8ifQ..PcSqE3rAZ9vILufr.MTVC6E92_4L_lprswHNQgAntGgVhxgc5pqZMWhdr6H2d_qUlhiaUqRkkrEpWXBcaWEmxElXj4KqM9mCDmHZV8qjzZ39BeGK80M_qIXoyOMoTB7HTSseWpT2o1xtLxilaJdQqG18vGpGm_4K7vwc7_Ywhgn8GdERLrvithoPXCdg2bkP_US4xORpr-uYf4cV5lmN8aLM2GxtiXmGcxFgCTBB_s5Lfz7oThclaOlPVOZ6jfWqpT0w3dbaUSzlAguf1sebJe4qyPUy50HVWRAn7ayheZdmZTOOrNO0B378KFJCq0D3K-DqvXPI-ZHhclwdzkiDhYDvVVeN_Z9M9FmtGkbL3fn6-tOWjOIGTAEUuNIyQgBI.dowQ_60y1d109swjhx9rtA",   "id_token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6ImNEbkFKVEhlVTVHTXg0Wmo2SDlraSJ9.eyJuaWNrbmFtZSI6ImZpcnN0dXNlciIsIm5hbWUiOiJmaXJzdHVzZXJAZm9vLmNvbSIsInBpY3RlcmUiOiJodHRwczovL3MuZ3JhdmdF0YXIuY29tLT2F2YXRhcy82OTM0OTBmNmY0NWl4MzZmYyYTRhN2FmYmFkMD9zPTQ4MCZyPXBnJmQ9aHR0cHMlM0ElMkYlMkZjZG4uYXV0aDAuY29tJTJGYXZhdGFycyUyRmZpLnBuZyIsInVwZGF0ZWRfYXQlOiIyMDIyLTA2LTA2VDIxOjI1OjQ4LjUyOVoicjE6ImZpcnN0dXNlckBmb28uY29tIiwiaWF0IjE6ImF1dGgwZDZyOWE4NTNiZTliNDk2MDA2OWYzNzg0MSIsImF1dGciOiJkVDR3VLaUJleG9uS3lDemNXQk5hMUTmMHJEMzdWZHZrIiwiaWF0IjoxNjU0NTUwNzQ4LjE4MjE0OjE2NTQ1ODY3NDh9.EgHtsoUhP_s6XknS6txFNIweM3BSc9rzMzaa2eSETqBmnlSWff3zK5zo53MQAeFJPDAMnUZKQ3nkQLz8Kqoi7lMETzhOLsKDBSdyTh_zVs2quAXean_ituTbpSyRsoL4Piu5wiOyascbv6qFQMLG3gTeLrZ8JvLGlf0yzrr7WiiAWbk8Nbcy-geM_uI8CGL0q4-yAY5DhN7ggd6TEHlyHj1-VhpYPJoFG_RkY_iG3AFyI9Pv5cAKnMOLJLlgccsZ4mbEPYljm3fyH3lh_gNhWlhqmttEjvVrECyaxA94ICssWiWdCRNPwKvgPqIvBuv2UBxzQFr1Q-MmAtLf54OHg",   "scope": "openid profile email address phone",   "expires_in": 86400,   "token_type": "Bearer" }</pre>

### Failure

<b>Status: 400 Bad Request</b>
<pre>{   "Error": "The request object is missing at least one of the required attributes" }</pre>

```
}
```

Status: 415 Unsupported Media Type

```
{  
  "Error": "Only JSON data can be accepted in request object"  
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "Only JSON data can be returned in response object"  
}
```

## Create a Boat

Creates a boat. All required attributes (*name*, *type*, and *length*) must be included in the request body. Extraneous attributes are ignored.

Endpoint Protected: Yes

POST /boats
-------------

### Request

Path Parameters

None

### Request Body

Required

### Request Headers

Required:

Content-Type: application/json

Accept: application/json

Authorization: Bearer {JWT}

### MIME Types for Request Body

application/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": "Sunrise",
  "type": "Sailboat",
  "length": 20
}
```

### Response

MIME Types for Response Body

Success: application/json

Failure: application/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, or any of the required attributes have invalid values.
Failure	415 Unsupported Media Type	If the request's 'Content-Type' header is not MIME type 'application/json'
Failure	406 Not Acceptable	If the request's 'Accept' header is not MIME type 'application/json'
Failure	401 Unauthorized	The user supplied JWT is either invalid or missing

## Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. The app sends back this value in the response body as shown in the example.
- A URL link to the representation of the resource is also sent back in the response body as "self".

### Success

Status: 201 Created

```
{
  "name": "Sunrise",
  "type": "Sailboat",
  "length": 20,
  "user": "auth0|629a853be9b4960069f37841",
  "loads": [],
  "self": "https://shipping-port-6543.ue.r.appspot.com/boats/5714080711049216",
  "id": 5714080711049216
}
```

### Failure

Status: 400 Bad Request

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

Status: 415 Unsupported Media Type

```
{
  "Error": "Only JSON data can be accepted in request object"
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "Only JSON data can be returned in response object"  
}
```

Status: 401 Unauthorized

```
{  
  "Error": "Missing or invalid JWT"  
}
```

## View User's Boats

Retrieves all boats tied to user corresponding to JWT

Endpoint Protected: Yes

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

### Request

#### Path Parameters

None

#### Request Body

None

#### Request Headers

Required:

Accept: application/json

Authorization: Bearer {JWT}

### Response

#### MIME Types for Response Body

Success: application/json

Failure: application/json

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	If the request's 'Accept' header is not MIME type 'application/json'
Failure	401 Unauthorized	The user supplied JWT is either invalid or missing

## Response Examples

- Returns empty array of boats if no boats have a relationship with the user entity

### Success

Status: 200 OK

```
{
  "boats": [
    {
      "length": 20,
      "loads": [
        {
          "id": 5066704988143616
        }
      ],
      "self": "https://shipping-port-6543.ue.r.appspot.com/loads/5066704988143616",
      "type": "Sailboat",
      "name": "Sunrise",
      "user": "auth0|629a853be9b4960069f37841",
      "id": 5714080711049216,
      "self": "https://shipping-port-6543.ue.r.appspot.com/boats/5714080711049216",
      "collection_count": "1 of 2"
    },
    {
      "type": "Catamaren",
      "name": "Sharky",
      "length": 35,
      "loads": [],
      "user": "auth0|629a853be9b4960069f37841",
      "id": 6279511075192832,
      "self": "https://shipping-port-6543.ue.r.appspot.com/boats/6279511075192832",
      "collection_count": "2 of 2"
    }
  ]
}
```

### Failure

Status: 406 Not Acceptable

```
{
  "Error": "Only JSON data can be returned in response object"
}
```



Status: 401 Unauthorized

```
{  
  "Error": "Missing or invalid JWT"  
}
```

## View a Boat

Views a boat with boat\_id.

Endpoint Protected: Yes

GET /boats/:boat_id
---------------------

### Request

#### Path Parameters

Name	Description
boat_id	ID of the boat

### Request Body

None

### Request Headers

Required:

Accept: application/json

Authorization: Bearer {JWT}

### Response

#### MIME Types for Response Body

Success: application/json

Failure: application/json

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	Invalid boat_id
Failure	406 Not Acceptable	If the request's 'Accept' header is not MIME type 'application/json'
Failure	401 Unauthorized	The user supplied JWT is either invalid or missing
Failure	403 Forbidden	Forbidden access to protected resource

### Response Examples

#### Success

Status: 200 OK
<pre>{   "user": "auth0 629a853be9b4960069f37841",   "length": 20,   "type": "Sailboat",   "name": "Sunrise",   "loads": [     {       "id": 5066704988143616,</pre>

```
"self": "https://shipping-port-6543.ue.r.appspot.com/loads/5066704988143616"
  },
  "id": 5714080711049216,
  "self": "https://shipping-port-6543.ue.r.appspot.com/boats/5714080711049216"
}
```

### Failure

Status: 404 Not Found

```
{
  "Error": "boat does not exist"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Only JSON data can be returned in response object"
}
```

Status: 401 Unauthorized

```
{
  "Error": "Missing or invalid JWT"
}
```

Status: 403 Forbidden

```
{
  "Error": "boat owned by different user"
}
```

## Delete a Boat

Deletes a specified boat.

Endpoint Protected: Yes

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

### Request

#### Path Parameters

Name	Description
boat_id	ID of the boat

#### Request Body

None

#### Request Headers

Required:

Accept: application/json

Authorization: Bearer {JWT}

### Response

#### MIME Types for Response Body

Success: None

Failure: application/json

#### Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	404 Not Found	Invalid boat_id
Failure	401 Unauthorized	The user supplied JWT is either invalid or missing
Failure	403 Forbidden	Forbidden access to protected resource

#### Response Examples

##### Success

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

##### Failure

Status: 404 Not Found
<pre>{   "Error": "boat does not exist" }</pre>
Status: 403 Forbidden
<pre>{   "Error": "boat owned by different user" }</pre>

Status: 401 Unauthorized

```
{  
  "Error": "Missing or invalid JWT"  
}
```

## Edit a Boat (PUT)

Updates a specified boat's attributes. All required attributes (*name*, *type*, and *length*) must be in the request body.

Endpoint Protected: Yes

PUT /boats/:boat_id
---------------------

### Request

#### Path Parameters

Name	Description
boat_id	ID of the boat

### Request Body

Required

### Request Headers

Required:

Content-Type: application/json

Accept: application/json

Authorization: Bearer {JWT}

### MIME Types for Request Body

application/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": "Beached",
  "type": "Sailboat",
  "length": 28
}
```

### Response

#### MIME Types for Response Body

Success: application/json

Failure: application/json

### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	Invalid boat_id
Failure	406 Not Acceptable	If the request's 'Accept' header is not MIME type 'application/json'
Failure	401 Unauthorized	The user supplied JWT is either invalid or missing
Failure	415 Unsupported Media Type	If the request's 'Content-Type' header is not 'application/json'
Failure	400 Bad Request	One of the required request body attributes is missing
Failure	403 Forbidden	Forbidden access to protected resource

### Response Examples

#### Success:

Status: 200 OK

```
{
  "id": 5714080711049216,
  "length": 28,
  "loads": [
    {
      "id": 5066704988143616,
      "self": "https://shipping-port-6543.ue.r.appspot.com/loads/5066704988143616"
    }
  ],
  "name": "Beached",
  "self": "https://shipping-port-6543.ue.r.appspot.com/boats/5714080711049216",
  "type": "Sailboat",
  "user": "auth0|629a853be9b4960069f37841"
}
```

#### Failure

Status: 404 Not Found

```
{
  "Error": "boat does not exist"
}
```

Status: 403 Forbidden

```
{
  "Error": "boat owned by different user"
}
```

Status: 415 Unsupported Media Type

```
{  
  "Error": "Only JSON data can be accepted in request object"  
}
```

Status: 400 Bad Request

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

Status: 406 Not Acceptable

```
{  
  "Error": "Only JSON data can be returned in response object"  
}
```

Status: 401 Unauthorized

```
{  
  "Error": "Missing or invalid JWT"  
}
```



## Edit a Boat (PATCH)

Updates a specified boat's attributes. At least one of the attributes (*name*, *type*, and *length*) must be in the request body.

Endpoint Protected: Yes

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

### Request

#### Path Parameters

Name	Description
boat_id	ID of the boat

### Request Body

Required

### Request Headers

Required:

Content-Type: application/json

Accept: application/json

Authorization: Bearer {JWT}

### MIME Types for Request Body

application/json

### Request JSON Attributes

Name	Description	Required?
name	The name of the boat.	No
type	The type of the boat. E.g., Sailboat, Catamaran, etc.	No
length	Length of the boat in feet.	No

- **Note:** At least one of the attributes needs to be in the request body

### Request Body Example

<pre>{   "name": "Sunset",   "length": 10 }</pre>
---------------------------------------------------

### Response

#### MIME Types for Response Body

Success: application/json

Failure: application/json

## Response Statuses

Outcome	Status Code	Notes
Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	Invalid boat_id
Failure	406 Not Acceptable	If the request's 'Accept' header is not MIME type 'application/json'
Failure	401 Unauthorized	The user supplied JWT is either invalid or missing
Failure	415 Unsupported Media Type	If the request's 'Content-Type' header is not 'application/json'
Failure	400 Bad Request	None of the required attributes are in the request body
Failure	403 Forbidden	Forbidden access to protected resource

## Response Examples

### Success:

Status: 200 OK
<pre>{   "id": 5714080711049216,   "length": 10,   "loads": [     {       "id": 5066704988143616,       "self": "https://shipping-port-6543.ue.r.appspot.com/loads/5066704988143616"     }   ],   "name": "Sunset",   "self": "https://shipping-port-6543.ue.r.appspot.com/boats/5714080711049216",   "type": "Sailboat",   "user": "auth0 629a853be9b4960069f37841" }</pre>

### Failure

Status: 404 Not Found
<pre>{   "Error": "boat does not exist" }</pre>
Status: 403 Forbidden
<pre>{   "Error": "boat owned by different user" }</pre>
Status: 415 Unsupported Media Type

```
{  
  "Error": "Only JSON data can be accepted in request object"  
}
```

Status: 400 Bad Request

```
{  
  "Error": "The request object does not contain at least one of the required attributes"  
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "Only JSON data can be returned in response object"  
}
```

Status: 401 Unauthorized

```
{  
  "Error": "Missing or invalid JWT"  
}
```

## Create a Load

Creates a load. All required attributes (name, quantity, and weight) must be included in the request body.

Endpoint Protected: No

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

### Request

Path Parameters

None

### Request Body

Required

### Request Headers

Required:

Content-Type: application/json

Accept: application/json

### MIME Types for Request Body

application/json

### Request JSON Attributes

Name	Description	Required?
name	The name of the boat.	Yes
quantity	Quantity of load	Yes
weight	Weight of load in pounds	Yes

### Request Body Example

```
{
  "name": "Oranges",
  "quantity": 100,
  "weight": 5
}
```

## Response

### MIME Types for Response Body

Success: application/json

Failure: application/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, or any of the required attributes have invalid values.

Failure	415 Unsupported Media Type	If the request's 'Content-Type' header is not MIME type 'application/json'
Failure	406 Not Acceptable	If the request's 'Accept' header is not MIME type 'application/json'

### Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. The app sends back this value in the response body as shown in the example.
- A URL link pointing to the representation of the resource is also sent back in the response body as "self".

### Success

Status: 201 Created

```
{
  "name": "oranges",
  "quantity": 100,
  "weight": 5,
  "carrier": null,
  "self": "https://shipping-port-6543.ue.r.appspot.com/loads/6272063266357248",
  "id": 6272063266357248
}
```

### Failure

Status: 400 Bad Request

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

Status: 415 Unsupported Media Type

```
{
  "Error": "Only JSON data can be accepted in request object"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Only JSON data can be returned in response object"
}
```

## View Loads

Retrieves all loads

Endpoint Protected: No

GET /loads
------------

### Request

Path Parameters

None

### Request Body

None

### Request Headers

Required:

Accept: application/json

### Response

MIME Types for Response Body

Success: application/json

Failure: application/json

### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	If the request's 'Accept' header is not MIME type 'application/json'

### Response Examples

#### Success

Status: 200 OK
<pre>{   "loads": [     {       "name": "oranges",       "weight": 5,       "quantity": 100,       "carrier": {         "id": 5714080711049216,         "self": "https://shipping-port-6543.ue.r.appspot.com/boats/5714080711049216"       },       "id": 5066704988143616,       "self": "https://shipping-port-6543.ue.r.appspot.com/loads/5066704988143616",       "collection_count": "1 of 2"     },   ], }</pre>

```
{
  "carrier": {
    "id": 5714080711049216,
    "self": "https://shipping-port-6543.ue.r.appspot.com/boats/5714080711049216"
  },
  "weight": 5,
  "quantity": 100,
  "name": "mangos",
  "id": 6272063266357248,
  "self": "https://shipping-port-6543.ue.r.appspot.com/loads/6272063266357248",
  "collection_count": "2 of 2"
}
```

#### *Failure*

Status: 406 Not Acceptable

```
{
  "Error": "Only JSON data can be returned in response object"
}
```

## View a Load

Views a load with load\_id.

Endpoint Protected: No

GET /loads/:load_id
---------------------

### Request

#### Path Parameters

Name	Description
load_id	ID of the load

### Request Body

None

### Request Headers

Required:

Accept: application/json

### Response

#### MIME Types for Response Body

Success: application/json

Failure: application/json

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	Invalid load_id
Failure	406 Not Acceptable	If the request's 'Accept' header is not MIME type 'application/json'

### Response Examples

#### Success

Status: 200 OK
<pre>{   "carrier": null,   "weight": 5,   "quantity": 100,   "name": "oranges",   "id": 5660308458700800,   "self": "https://shipping-port-6543.ue.r.appspot.com/loads/5660308458700800" }</pre>

#### Failure

Status: 404 Not Found
-----------------------



```
{  
  "Error": "load does not exist"  
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "Only JSON data can be returned in response object"  
}
```

## Delete a Load

Deletes a specified load.

Endpoint Protected: No

DELETE /loads/:load_id
------------------------

### Request

#### Path Parameters

Name	Description
load_id	ID of the load

### Request Body

None

### Request Headers

Required:

Accept: application/json

### Response

#### MIME Types for Response Body

Success: None

Failure: application/json

### Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	404 Not Found	Invalid load_id

### Response Examples

#### Success

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

#### Failure

Status: 404 Not Found  <pre>{   "Error": "load does not exist" }</pre>
--------------------------------------------------------------------------------------

## Edit a Load (PUT)

Updates a specified load's attributes. All required attributes (*name*, *quantity*, and *weight*) must be in the request body.

Endpoint Protected: No

PUT /loads/:loads_id
----------------------

### Request

#### Path Parameters

Name	Description
load_id	ID of the load

### Request Body

Required

### Request Headers

Required:

Content-Type: application/json

Accept: application/json

### MIME Types for Request Body

application/json

### Request JSON Attributes

Name	Description	Required?
name	The name of the load	Yes
quantity	Quantity of the load	Yes
length	Weight of the load	Yes

### Request Body Example

```
{
  "name": "Mangos",
  "quantity": 100,
  "weight": 5
}
```

### Response

#### MIME Types for Response Body

Success: application/json

Failure: application/json

## Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	Invalid load_id
Failure	406 Not Acceptable	If the request's 'Accept' header is not MIME type 'application/json'
Failure	415 Unsupported Media Type	If the request's 'Content-Type' header is not 'application/json'
Failure	400 Bad Request	One of the required request body attributes is missing

## Response Examples

### Success:

Status: 200 OK

```
{
  "carrier": {
    "id": 5714080711049216,
    "self": "https://shipping-port-6543.ue.r.appspot.com/boats/5714080711049216"
  },
  "id": 6272063266357248,
  "name": "mangos",
  "quantity": 100,
  "self": "https://shipping-port-6543.ue.r.appspot.com/loads/6272063266357248",
  "weight": 5
}
```

### Failure

Status: 404 Not Found

```
{
  "Error": "load does not exist"
}
```

Status: 415 Unsupported Media Type

```
{
  "Error": "Only JSON data can be accepted in request object"
}
```

Status: 400 Bad Request

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

Status: 406 Not Acceptable

```
{
```

```
"Error": "Only JSON data can be returned in response object"
```

```
}
```

## Edit a Load (PATCH)

Updates a specified load's attributes. At least one of the attributes (*name*, *quantity*, and *weight*) must be in the request body.

Endpoint Protected: No

PATCH /loads/:load_id
-----------------------

### Request

#### Path Parameters

Name	Description
load_id	ID of the load

### Request Body

Required

### Request Headers

Required:

Content-Type: application/json

Accept: application/json

### MIME Types for Request Body

application/json

### Request JSON Attributes

Name	Description	Required?
name	The name of the load.	No
quantity	The quantity of the load	No
length	Weight of the load	No

- **Note:** At least one of the attributes needs to be in the request body

### Request Body Example

```
{
  "quantity": 1000,
  "weight": 50
}
```

### Response

#### MIME Types for Response Body

Success: application/json

Failure: application/json

### Response Statuses

Outcome	Status Code	Notes
---------	-------------	-------

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	Invalid load_id
Failure	406 Not Acceptable	If the request's 'Accept' header is not MIME type 'application/json'
Failure	415 Unsupported Media Type	If the request's 'Content-Type' header is not 'application/json'
Failure	400 Bad Request	None of the required attributes are in the request body

## Response Examples

### Success:

Status: 200 OK

```
{
  "carrier": {
    "id": 5714080711049216,
    "self": "https://shipping-port-6543.ue.r.appspot.com/boats/5714080711049216"
  },
  "id": 6272063266357248,
  "name": "mangos",
  "quantity": 1000,
  "self": "https://shipping-port-6543.ue.r.appspot.com/loads/6272063266357248",
  "weight": 50
}
```

### Failure

Status: 404 Not Found

```
{
  "Error": "boat does not exist"
}
```

Status: 415 Unsupported Media Type

```
{
  "Error": "Only JSON data can be accepted in request object"
}
```

Status: 400 Bad Request

```
{
  "Error": "The request object does not contain at least one of the required attributes"
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "Only JSON data can be returned in response object"  
}
```



## Assign a Load to a Boat

Assigns a specified load to a specified boat.

Endpoint Protected: Yes

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

Required:

Accept: application/json

Authorization: Bearer {JWT}

### Response

#### MIME Types for Response Body

Success: None

Failure: application/json

#### Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	404 Not Found	Invalid boat_id and or load_id
Failure	401 Unauthorized	The user supplied JWT is either invalid or missing
Failure	403 Forbidden	Forbidden access to protected resource

### Response Examples

#### Success:

Status: 204 No Content

#### Failure

Status: 404 Not Found

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

Status: 403 Forbidden

```
{  
  "Error": "boat owned by different user"  
}
```

Status: 401 Unauthorized

```
{  
  "Error": "Missing or invalid JWT"  
}
```

## Remove a Load from a Boat

Removes a specified load from a specified boat.

Endpoint Protected: Yes

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 Body

None

### Request Headers

Required:

Accept: application/json

Authorization: Bearer {JWT}

### Response

#### MIME Types for Response Body

Success: None

Failure: application/json

#### Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	404 Not Found	Invalid boat_id and or load_id
Failure	401 Unauthorized	The user supplied JWT is either invalid or missing
Failure	403 Forbidden	Forbidden access to protected resource

### Response Examples

#### Success:

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

#### Failure

Status: 404 Not Found
<pre>{   "Error": "The specified boat and/or load does not exist" }</pre>

}
Status: 403 Forbidden  { "Error": "boat owned by different user" }
Status: 401 Unauthorized  { "Error": "Missing or invalid JWT" }