

# Portfolio Project

CS 493: Cloud Application Development

Spring 2022

Oregon State University

Last Update: June 05, 11:26 pm Central

## Application & Account Creation/Login:

<https://portfolio-tristan-harville.uk.r.appspot.com/>

Change log .....	1
Data Model .....	2
View all Users.....	4
Create a Boat.....	6
View a Boat .....	8
View all Boats.....	10
View all User Boats.....	12
Edit a Boat.....	14
Update a Boat .....	16
Delete a Boat.....	18
Create a Load .....	20
View a Load.....	22
View all Loads.....	24
Edit a Load.....	26
Update a Load .....	28
Delete a Load .....	30
Assign Load to Boat .....	31
Remove Load from Boat .....	32

## Change log

Version	Change	Date
1.0	Initial version.	April 25, 2022

## Data Model

The app stores three kinds of entities in Datastore, Boats, Loads, and Users. All properties are required in order for the entity to exist.

### Boats

Property	Data Type	Valid Values
id	Integer	The id of the boat. This is automatically generated by DataStore and should not be specified.
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	This is an array containing load objects that the boat currently has loaded on it. These load objects can only be placed on one boat. If the boat is carrying no loads, the array will be empty.
Owner	String	The owner of the array, uses the sub property of OAuth as the identifier

### Loads

Property	Data Type	Valid Values
id	Integer	The id of the load. This is automatically generated by DataStore and should not be specified.
volume	Integer	How much space the load takes up on the boat.
carrier	Object	If a boat is currently carrying this load, an object containing information about the boat will be stored here. If no boat is currently carrying this load, then the property will contain "null".
item	String	A description of what the load is.
creation_date	String	That date that this load was created.

### Users

Property	Data Type	Valid Values
id	Integer	The id of the user. This is automatically generated by DataStore and should not be specified.
name	String	Name of the user.
Sub	String	The sub property of the OAuth login that is generated when a user logs onto the application.

### Non-user Entity Relationship

This application contains one non-user relationship between the Loads entity and the Boats entity. This relationship is a one-to-many relationship, where a load can only be associated with one boat but a boat can be associated with many loads. A boat may exist without any loads. A load may exist without a carrier, which is a boat.

## User Entity Model

This application also supports a user entity. Users are authenticated through the use of auth0 and authorized using a JWT. The unique identifier for a user is the 'sub' property of the JWT. If a user were to make a request to the API for protected endpoints, the user would need to provide the JWT provided upon logging into the application. The application does not require the use of a user\_id.

A relationship between a user entity and a non-user entity is also supported by this application. This relationship is a one-to-many relationship between users and boats. A user may have many boats. A boat may have only one user, who is identified as the boat's 'owner'. A user may exist without owning any boats. A boat may not exist without an owner.

## View all Users

Get all the users that have logged onto the application.

GET /users

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	The MIME type requested is not supported

Response Examples

*Success*

Status: 200 OK

```
{
  "users":
  [
    {
      "id": 123,
      "name": "David Johnson",
      "sub": auth0|232tnen11231nasn,
    },
    {
      "id": 333,
      "name": "Chris Brown",
      "sub": auth0|545ggfsd33sn,
    },
    {
      "id": 890,
      "name": "Strange Guy",
      "owner": auth0|9578eunfd12384,
    }
  ]
}
```

*Failure*

Status: 406 Not Acceptable

```
{  
  "Error": "Not Acceptable MIME type"  
}
```

## Create a Boat

Allows you to create a new boat.

POST /boats

Request

Path Parameters

None

Request Header

Bearer Token

Request Body

Required

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 400 status code will be returned.  It is assumed that the properties datatype input will be as stated in the data model for the boat
Failure	401 Unauthorized	The user is not logged in and therefore does not have permission to create boats
Failure	406 Not Acceptable	The MIME type requested is not supported

## Response Examples

### Success

Status: 201 Created

```
{
  "id": 123,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "loads": [],
  "owner": auth0|232tnen11231nasn,
  "self": 28,
}
```

### Failure

Status: 400 Bad Request

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

Status: 401 Unauthorized

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

Status: 406 Not Acceptable

```
{
  "Error": "Not Acceptable MIME type"
}
```

## View a Boat

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

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No boat with this boat_id exists
Failure	406 Not Acceptable	The MIME type requested is not supported

#### Response Examples

##### *Success, no loads*

Status: 200 OK

```
{
  "id": 123,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "loads": [],
  "owner": auth0|232tnen11231nasn,
  "self": "https://hw4-tristan-harville.uk.r.appspot.com/boats/123",
}
```

##### *Success, with loads*

Status: 200 OK

```
{
  "id": 333,
  "name": "Adventure",
  "type": "Sailboat",
  "length": 50
  "loads": [
    {
      "self": "https://hw4-tristan-harville.uk.r.appspot.com/loads/456",
      "id": "456"
    }
  ]
}
```



```
{
  "self": "https://hw4-tristan-harville.uk.r.appspot.com/loads/777",
  "id": "777"
},
{
  "owner": "auth0|232tnen11231nasn",
  "self": "https://hw4-tristan-harville.uk.r.appspot.com/boats/333",
}
```

#### *Failure*

Status: 404 Not Found

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

Status: 406 Not Acceptable

```
{
  "Error": "Not Acceptable MIME type"
}
```

## View all Boats

Get all the boats in DataStore. This method utilizes pagination with the limit and offset parameters. These parameters are optional. When they are not specified, a limit of 5 and offset of 0 are used.

GET /boats

### Request

#### Path Parameters

Parameter	Description	Required?
Limit	How many load objects are to be returned per result	No
Offset	The starting point in the load dataset to start your results	No

#### Request Body

None

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	The MIME type requested is not supported

### Response Examples

#### Success

Status: 200 OK

```
[
{
  "id": 123,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "loads": [],
  "owner": "auth0|232tnen11231nasn",
  "self": "https://hw4-tristan-harville.uk.r.appspot.com/boats/123",
},
{
  "id": 333,
  "name": "Adventure",
  "type": "Sailboat",
  "length": 50
  "loads": [
    {"self": "https://hw4-tristan-harville.uk.r.appspot.com/loads/456",
      "id": "456"},
    {"self": "https://hw4-tristan-harville.uk.r.appspot.com/loads/777",
      "id": "777"}
  ],
}
```

```
"owner": auth0|545ggfsd33sn,  
"self": "https://hw4-tristan-harville.uk.r.appspot.com/boats/333",  
},  
{  
  "id": 890,  
  "name": "Kodiak",  
  "type": "Canoe",  
  "length": 12,  
  "loads": [],  
  "owner": auth0|9578eunfd12384,  
  "self": "https://hw4-tristan-harville.uk.r.appspot.com/boats/890",  
}  
]
```

#### *Failure*

Status: 406 Not Acceptable

```
{  
  "Error": "Not Acceptable MIME type"  
}
```

## View all User Boats

Get all the boats in DataStore that are associated with a user and is protected. This method utilizes pagination with the limit and offset parameters. These parameters are optional. When they are not specified, a limit of 5 and offset of 0 are used.

GET /boats

### Request

#### Path Parameters

Parameter	Description	Required?
Limit	How many load objects are to be returned per result	No
Offset	The starting point in the load dataset to start your results	No

#### Request Header

Bearer token

#### Request Body

None

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	The user is not logged in, and an error will throw and the default function of view all boats will be executed
Failure	406 Not Acceptable	The MIME type requested is not supported

### Response Examples

#### Success

Status: 200 OK
<pre>[ {   "id": 123,   "name": "Sea Witch",   "type": "Catamaran",   "length": 28,   "loads": [],   "owner": auth0 232tnen11231nasn,   "self": "https://hw4-tristan-harville.uk.r.appspot.com/boats/123", }, {   "id": 333,   "name": "Adventure",   "type": "Sailboat",</pre>

```
"length": 50
"loads": [
  {"self": "https://hw4-tristan-harville.uk.r.appspot.com/loads/456",
   "id": "456"},
  {"self": "https://hw4-tristan-harville.uk.r.appspot.com/loads/777",
   "id": "777"}
],
"owner": auth0|232tnen11231nasn,
"self": "https://hw4-tristan-harville.uk.r.appspot.com/boats/333",
},
{
  "id": 890,
  "name": "Kodiak",
  "type": "Canoe",
  "length": 12,
  "loads": [],
  "owner": auth0|232tnen11231nasn,
  "self": "https://hw4-tristan-harville.uk.r.appspot.com/boats/890",
}
]
```

#### *Failure*

Status: 406 Not Acceptable

```
{
  "Error": "Not Acceptable MIME type"
}
```

## Edit a Boat

Allows the user to edit a boat. This is a protected endpoint and the user must be logged in and authenticated to edit a boat.

PATCH /boats/:boat\_id

### Request

#### Path Parameters

Parameter	Description	Required?
boat_id	ID of the boat	Yes

#### Request Header

Bearer token

#### Request Body

Required

#### Request Body Format

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

#### Request Body Example

```
{
  "type": "Catamaran"
}
```

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	The user did not enter any attributes to be edited
Failure	401 Unauthorized	The user is not logged in
Failure	403 Forbidden	The user is trying to edit a boat owned by another user
Failure	404 Not Found	The boat_id was not found in the database
Failure	406 Not Acceptable	The MIME type requested is not supported

## Response Examples

### Success

Status: 200 OK

```
{
  "id": 123,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "loads": [],
  "owner": "auth0|232tnen11231nasn",
  "self": "https://hw4-tristan-harville.uk.r.appspot.com/boats/123",
}
```

### Failure

Status: 400 Bad Request

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

Status: 401 Unauthorized

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

Status: 403 Forbidden

```
{
  "Error": "Cannot update this boat since it is owned by another user"
}
```

Status: 404 Not Found

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

Status: 406 Not Acceptable

```
{
  "Error": "Not Acceptable MIME type"
}
```

## Update a Boat

Allows the user to update a boat. This is a protected endpoint and the user must be logged in and authenticated to update a boat.

PUT /boats/:boat\_id

### Request

#### Path Parameters

Parameter	Description	Required?
boat_id	ID of the boat	Yes

#### Request Header

Bearer token

#### Request Body

Required

#### 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	200 OK	
Failure	400 Bad Request	The user did not enter all required attributes to be updated
Failure	401 Unauthorized	The user is not logged in
Failure	403 Forbidden	The user is trying to update a boat owned by another user
Failure	404 Not Found	The boat_id was not found in the database
Failure	406 Not Acceptable	The MIME type requested is not supported



## Response Examples

### Success

Status: 200 OK

```
{
  "id": 123,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "loads": [],
  "owner": auth0|232tnen11231nasn,
  "self": "https://hw4-tristan-harville.uk.r.appspot.com/boats/123",
}
```

### Failure

Status: 400 Bad Request

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

Status: 401 Unauthorized

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

Status: 403 Forbidden

```
{
  "Error": "Cannot update this boat since it is owned by another user"
}
```

Status: 404 Not Found

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

Status: 406 Not Acceptable

```
{
  "Error": "Not Acceptable MIME type"
}
```

## Delete a Boat

Allows you to delete a boat. This is a protected endpoint and the user must be logged into the application to have authorization to delete a boat. Note that if the boat is currently carrying loads, deleting the boat will mark each load as not being carried (i.e. the 'carrier' property for the load will have a value of 'null').

DELETE /boats/:boat\_id

### Request

#### Path Parameters

Name	Description
boat_id	ID of the boat

### Request Header

Bearer token

### 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	The user is not logged in
Failure	403 Forbidden	The user is trying to delete a boat owned by another user
Failure	404 Not Found	The boat_id was not found in the database

### Response Examples

#### Success

Status: 204 No Content

#### Failure

Status: 401 Unauthorized

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

Status: 403 Forbidden

```
{  
"Error": "Cannot delete this boat since it is owned by another user"  
}
```

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	The description of what the load is.	Yes
creation_date	Date that this load was created	Yes

Request Body Example

```
{
  "volume": 12,
  "item": "Milk Cartons",
  "creation_date": "09/16/2020"
}
```

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 load will not be created, and 400 status code will be returned  It is assumed that the properties datatype input will be as stated in the data model for the load.
Failure	406	

## Response Examples

### *Success*

Status: 201 Created

```
{
  "id": 123,
  "volume": 12,
  "carrier": null,
  "item": "Milk Cartons",
  "creation_date": "09/16/2020",
  "self": "https://hw4-tristan-harville.uk.r.appspot.com/loads/123",
}
```

### *Failure*

Status: 400 Bad Request

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

Status: 406 Not Acceptable

```
{
  "Error": "Not Acceptable MIME type"
}
```

## View 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
Failure	406 Not Acceptable	The MIME type requested is not supported

Response Examples

*Success, no carrier*

Status: 200 OK

```
{
  "id": 123,
  "volume": 12,
  "carrier": null,
  "item": "Milk Cartons",
  "creation_date": "09/16/2020",
  "self": "https://hw4-tristan-harville.uk.r.appspot.com/loads/123",
}
```

*Success, with carrier*

Status: 200 OK

```
{
  "id": 456,
  "volume": 44,
  "carrier": {
    "self": "https://hw4-tristan-harville.uk.r.appspot.com/boats/777",
    "id": "777",
    "name": "Sea Witch"},
  "item": "Bowling Balls",
}
```

```
"creation_date": "03/22/2021",  
"self": "https://hw4-tristan-harville.uk.r.appspot.com/loads/456",  
}
```

#### *Failure*

Status: 404 Not Found

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

Status: 406 Not Acceptable

```
{  
"Error": "Not Acceptable MIME type"  
}
```

## View all Loads

Get all the loads in DataStore. This method utilizes pagination with the limit and offset parameters. These parameters are optional. When they are not specified, a limit of 5 and offset of 0 are used.

GET /loads

### Request

#### Path Parameters

Parameter	Description	Required?
Limit	How many load objects are to be returned per result	No
Offset	The starting point in the load dataset to start your results	No

#### Request Body

None

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	The MIME type requested is not supported

### Response Examples

#### Success

Status: 200 OK

```
[
{
  "id": 123,
  "volume": 12,
  "carrier": null,
  "item": "Milk Cartons",
  "creation_date": "09/16/2020",
  "self": "https://hw4-tristan-harville.uk.r.appspot.com/loads/123",
},
{
  "id": 456,
  "volume": 44,
  "carrier": {
    "self": "https://hw4-tristan-harville.uk.r.appspot.com/boats/777",
    "id": "777",
    "name": "Sea Witch"},
  "item": "Bowling Balls",
  "creation_date": "03/22/2021",
  "self": "https://hw4-tristan-harville.uk.r.appspot.com/loads/456",
},
{

```



```
"id": 778,  
"volume": 120,  
"carrier": null,  
"item": "Tractors",  
"creation_date": "12/25/2021",  
"self": "https://hw4-tristan-harville.uk.r.appspot.com/loads/778",  
}  
]
```

#### *Failure*

Status: 406 Not Acceptable

```
{  
"Error": "Not Acceptable MIME type"  
}
```

## Edit a Load

Allows the user to edit a load.

PATCH /loads/:load\_id

### Request

#### Path Parameters

Parameter	Description	Required?
load_id	ID of the load	Yes

#### Request Body

Required

#### Request Body Format

JSON

#### Request JSON Attributes

Name	Description	Required?
volume	The volume of the load.	No
item	The description of what the load is.	No
creation_date	Date that this load was created	No

#### Request Body Example

```
{
  "item": "Milk Duds",
}
```

## Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	The user did not enter any attributes to be edited
Failure	404 Not Found	The load_id was not found in the database
Failure	406 Not Acceptable	The MIME type requested is not supported

#### Response Examples

##### Success

Status: 200 OK

```
{
  "id": 123,
  "volume": 12,
  "carrier": null,
}
```

```
"item": "Milk Duds",  
"creation_date": "09/16/2020",  
"self": "https://hw4-tristan-harville.uk.r.appspot.com/loads/123",  
}
```

#### *Failure*

Status: 400 Bad Request

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

Status: 404 Not Found

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

Status: 406 Not Acceptable

```
{  
"Error": "Not Acceptable MIME type"  
}
```

## Update a Load

Allows the user to update a load.

PATCH /loads/:load\_id

### Request

#### Path Parameters

Parameter	Description	Required?
load_id	ID of the load	Yes

#### Request Body

Required

#### Request Body Format

JSON

#### Request JSON Attributes

Name	Description	Required?
volume	The volume of the load.	Yes
item	The description of what the load is.	Yes
creation_date	Date that this load was created	Yes

#### Request Body Example

```
{
  "volume": 2,
  "item": "Milk Duds",
  "creation_date": "10/29/2020",
}
```

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	The user did not enter all the required attributes to update
Failure	404 Not Found	The load_id was not found in the database
Failure	406 Not Acceptable	The MIME type requested is not supported

#### Response Examples

##### Success

Status: 200 OK

```
{
```

```
"id": 123,  
"volume": 2,  
"carrier": null,  
"item": "Milk Duds",  
"creation_date": "10/29/2020",  
"self": "https://hw4-tristan-harville.uk.r.appspot.com/loads/123",  
}
```

#### *Failure*

Status: 400 Bad Request

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

Status: 404 Not Found

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

Status: 406 Not Acceptable

```
{  
"Error": "Not Acceptable MIME type"  
}
```

## Delete a Load

Allows you to delete a load. Note that if the load is currently being carried by a boat, deleting the load will remove the load from the boat it has been placed on (i.e. the 'loads' property for the boat will remove the load from the value array).

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"
}
```

## Assign Load to Boat

A load is assigned to be carried by a boat. A boat can be assigned many loads. A load may only be assigned to one 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

### 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 not currently carried by another boat.
Failure	403 Forbidden	The load is already being carried by another boat.
Failure	404 Not Found	The specified boat and/or load does not exist.

### Response Examples

#### Success

Status: 204 No Content

#### 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"
}
```

## Remove Load from Boat

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 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 currently carried by the boat.
Failure	404 Not Found	The specified boat and/or load does not exist, or the load is not on the boat specified.

### Response Examples

#### Success

Status: 204 No Content

#### Failure

Status: 404 Not Found

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

Status: 404 Not Found

```
{
  "Error": "boat_id or load_id does not exist"
}
```