

Boat Tracker

Author: Robert Elsom

Change log.....	1
Descriptions	2
Create a Boat	3
Get a Boat	5
List all Boats	6
Edit a Boat - Patch.....	7
Edit a Boat – Put.....	10
Delete a Boat.....	12
Create a Load	13
Get a Load	14
List all Loads	15
Edit a Load - Patch.....	17
Edit a Load – Put	19
Delete a Load	21
Add Load to a Boat.....	22
Remove a Load from a Boat.....	23
View all Loads for a Boat.....	24
View all Boats belonging to a User	26

Change log

Page	Change	Date
	Initial version.	M, Dec 9.

Descriptions

Entities

There are two non-user entities. The boat entity that has a name, type and length property and a load entity that has weight, content, and delivery date properties. Each boat can hold many loads, but one load can only be assigned to one boat at a time. The loads for a boat are tracked under a property called “loads” that contains a list of all of the load ids assigned to the boat. A load also tracks what boat it is assigned to using a ‘carrier’ property that contains the boat id it is assigned to. The boat also tracks which user created it and has rights to it under the ‘owner’ property, which contains a unique identifier for the user that created the boat and is the only user that can edit or delete the boat or add and remove loads from that boat.

User Entity

The user entity is created using OAUTH and a google sign on. The unique identifier stored in the boats ‘owner’ property and is taken from the ‘sub’ property belonging to their JWT token. If a request is being sent that needs authorization, the user must set an Authorization header to ‘Bearer <jwt_token>’ where jwt_token is the token displayed after creating an account at the root url. The application then uses this token, gets the ‘sub’ value, and checks against the stored ‘owner’ of the boat to make sure that the current token is the correct value to let the user edit or delete the boat. The relationship between the user and non-user entity is a user is the owner of as many boats as they want and is the only one who has the right to edit or delete said boats, but each boat only has one owner, the one who created it.

Note: To test unauthorized access in the postman suite, set the jwt_token 2 environment variable to a valid token that belongs to another user.

Input Verification:

Below are the verification checks done one request bodies:

Name	Entity	Allowable Characters	Max Length
name	Boat	Letters, numbers, and spaces	50
type	Boat	Letters and space.	30
length	Boat	Numbers only	6
Weight	Load	Numbers only	20
Content	Load	Letters, numbers, and spaces	50
Delivery_date	Load	Numbers and ‘-’, ‘/’, ‘\’ characters	11

AUTHORIZATION REQUIRED MEANS THAT THE JWT TOKEN IS REQUIRED TO USE THE ENDPOINTS FUNCTIONALITY. THIS MEANS THE TOKEN ENVIRONMENT VARIABLE MUST BE SET IN POSTMAN BEFORE SENDING ANY REQUEST. TESTS IN THE POSTMAN SUITE WILL ALSO FAIL IF “jwt_token 2” ENVIRONMENT VARIABLE IS NOT SET TO A VALID JWT TOKEN FROM A DIFFERENT EMAIL ADDRESS THAN THE “TOKEN” VARIABLE.

Create a Boat

Allows you to create a new boat.

POST /boats

Request

Authorization

Required

Request Parameters

None

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	If the request is missing any of the 3 required attributes, the boat must not be created, and 400 status code must be returned.
Failure	401 Missing Credentials	If there are invalid or missing jwt credentials, a 401 status is returned
Failure	406 Invalid Request	If the mime-type that is requested to be returned is not application/json or the request header or the request body did not have an application/json content-type

Response Examples

Success

Status: 201 Created

```
{
  "id": "abc123",
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "owner": "12341234",
  "loads": []
  "self": "https://<your-app>/boats/abc123"
}
```

Failure

Status: 400 Bad Request

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

Failure

Status: 401 Missing Credentials

```
{
  "Error": "The JWT credentials are invalid or missing"
}
```

Failure

Status: 406 Invalid Request

```
{
  "Error": "Request Header not Acceptable"
}
```

Get a Boat

Allows you to get an existing boat

GET /boats/:boat_id

Request

Authorization

None

Request Parameters

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

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No boat with this boat_id exists
Failure	406 Invalid Request	If the mime-type that is requested to be returned is not application/json.

Response Examples

Success

<p>Status: 200 OK</p> <pre>{ "id": "abc123", "name": "Sea Witch", "type": "Catamaran", "length": 28, "owner": "12341234", "loads": [], "self": "https://<your-app>/boats/abc123" }</pre>
--

Failure

<p>Status: 404 Not Found</p> <pre>{ "Error": "No boat with this boat_id exists" }</pre>

Failure

<p>Status: 406 Invalid Request</p>

```
{
  "Error": " Request Header not Acceptable"
}
```

List all Boats

List all the boats.

GET /boats

Request

Authorization

None

Request Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Invalid Request	If the mime-type that is requested to be returned is not application/json.

Response Examples

Success

Status: 200 OK

```
[
{
  "id": "abc123",
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "owner": "12398349598273",
  "loads": [],
  "self": "https://<your-app>/boats/abc123"
},
```

```
{
  "id": "def456",
  "name": "Adventure",
  "type": "Sailboat",
  "owner": "12398349598273",
  "loads": [],
  "length": 50,
  "self": "https://<your-app>/boats/def456"
},
{
  "id": "xyz123",
  "name": "Hocus Pocus",
  "type": "Sailboat",
  "owner": "12398349598273",
  "loads": [
    {
      "id": "231",
      "self": "https://<your-app>/loads/231"
    }
  ],
  "length": 100,
  "self": "https://<your-app>/boats/xyz123"
}
]
```

Failure

Status: 406 Invalid Request

```
{
  "Error": "Request Header not Acceptable"
}
```

Edit a Boat - Patch

Allows you to edit a boat. To edit one or more attributes, use put. To edit all attributes use PATCH

PATCH /boats/:boat_id

Request

Authorization

Required

Request Parameters

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

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": 99,
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	<p>If the request is missing any of the 3 required attributes, the boat must not be updated, 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 other than the ones that are listed).</p>
Failure	404 Not Found	No boat with this boat_id exists
Failure	406 Invalid Request	If the mime-type that is requested to be returned is not application/json.
Failure	401 Invalid Credentials	JWT credentials are missing or invalid
Failure	403 Unauthorized	The boat belongs to someone else

Response Examples

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

Success

Status: 200 OK

```
{
```



```
{
  "id": "abc123",
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "owner": "12341234",
  "loads": [],
  "self": "https://<your-app>/boats/abc123"
}
```

Failure

Status: 400 Bad Request

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

Status: 404 Not Found

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

Failure

Status: 406 Invalid Request

```
{
  "Error": "Request Header not Acceptable"
}
```

Failure

Status: 401 Invalid Credentials

```
{
  "Error": "invalid or missing credentials"
}
```

Failure

Status: 403 Unauthorized

```
{
  "Error": "The boat belongs to someone else"
}
```

Edit a Boat – Put

Used to edit one or more attributes, but not necessarily all attributes.

PUT /boats/:boat_id

Request

Authorization

Required

Request Parameters

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

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

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

Request Body Example

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

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	303 OK	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, the boat must not be updated, and 400 status code must be returned.
Failure	404 Not Found	No boat with this boat_id exists
Failure	406 Invalid Request	If the mime-type that is requested to be returned is not application/json.
Failure	401 Invalid Credentials	JWT credentials are missing or invalid
Failure	403 Unauthorized	The boat belongs to someone else

Response Examples

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

Success

Status: 303 OK

```
{
  "id": "abc123",
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "owner": "12341234",
  "loads": [],
  "self": "https://<your-app>/boats/abc123"
}
```

Failure

Status: 400 Bad Request

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

Status: 404 Not Found

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

Failure

Status: 406 Invalid Request

```
{
  "Error": "Request Header not Acceptable"
}
```

Failure

Status: 401 Invalid Credentials

```
{
  "Error": "invalid or missing credentials"
}
```

Failure

Status: 403 Unauthorized

```
{
  "Error": "The boat belongs to someone else"
}
```

Delete a Boat

Allows you to delete a boat. Note that if the boat is currently shipping a load, the loads carrier is set to null.

DELETE /boats/:boat_id

Request

Authorization

Required

Request Parameters

None

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	404 Not Found	No boat with this boat_id exists
Failure	401 Invalid Credentials	JWT credentials are missing or invalid
Failure	403 Unauthorized	The boat belongs to someone else

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found

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

Failure

Status: 401 Invalid Credentials

```
{
  "Error": "invalid or missing credentials"
}
```

Failure

Status: 403 Unauthorized

```
{
  "Error": "The boat belongs to someone else"
}
```

Create a Load

Allows you to create a new load.

POST /loads

Request

Authorization

None

Request Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
Weight	Integer	The weight of the load	Yes
Content	String	The content of the load	Yes
Delivery_date	String	The expected date of delivery for the load	Yes

Request Body Example

```
{
  "weight": 1,
  "content": "Legos",
  "delivery_date": "1/1/2020"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	

Failure	406 Invalid Request	If the mime-type that is requested to be returned is not application/json or the request header or the request body did not have an application/json content-type
---------	---------------------	---

Response Examples

Success

<p>Status: 201 Created</p> <pre>{ "id": 5093108584808448, "weight": 1, "carrier": null, "content": "Legos", "delivery_date": "1/1/2020", "self": "https://elsomr-final.appspot.com/loads/5093108584808448" }</pre>
--

Failure

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

Failure

<p>Status: 406 Invalid Request</p> <pre>{ "Error": " Request Header not Acceptable" }</pre>

Get a Load

Allows you to get an existing load.

GET /loads/:load_id

Request

Authorization

None

Request Parameters

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

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 Invalid Request	If the mime-type that is requested to be returned is not application/json or the request header or the request body did not have an application/json content-type

Response Examples

Success

```
Status: 200 OK

{
  "id": 5093108584808448,
  "weight": 1,
  "carrier": null,
  "content": "Legos",
  "delivery_date": "1/1/2020",
  "self": "https://elsomr-final.appspot.com/loads/5093108584808448"
}
```

Failure

```
Status: 404 Not Found

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

Failure

```
Status: 406 Invalid Request

{
  "Error": " Request Header not Acceptable"
}
```

List all Loads

List all the loads.

GET /loads

Request

Authorization

None

Request Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Invalid Request	If the mime-type that is requested to be returned is not application/json or the request header or the request body did not have an application/json content-type

Response Examples

Success

Status: 200 OK

```
[
  {
    "id": 5093108584808448,
    "content": "Legos",
    "carrier": null,
    "weight": 1,
    "delivery_date": "1/1/2020",
    "self": "https://elsomr-final.appspot.com/loads/5093108584808448",
    "next": "https://elsomr-final.appspot.com/loads?limit=5&offset=5"
  },
  {
    "id": 5099055302574080,
    "weight": 25,
    "carrier": null,
    "delivery_date": "4/4/2020",
    "content": "Chips",
    "self": "https://elsomr-final.appspot.com/loads/5099055302574080",
    "next": "https://elsomr-final.appspot.com/loads?limit=5&offset=5"
  },
]
```


Failure

Status: 406 Invalid Request

```
{
  "Error": "Request Header not Acceptable"
}
```

Edit a Load - Patch

Allows you to edit a load. To edit one or more attributes, use put. To edit all attributes use PATCH

PATCH /loads/:load_id

Request

Authorization

None

Request Parameters

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

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
Weight	String	The weight of the load	Yes
Content	String	The content of the load	Yes
delivery_date	Integer	The expected date of delivery of the load	Yes

Request Body Example

```
{
  "weight": 1,
  "content": "Legos",
  "delivery_date": "1/1/2020"
}
```

Response

Response Body Format

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, the load must not be updated, and 400 status code must be returned.
Failure	404 Not Found	No load with this load_id exists
Failure	406 Invalid Request	If the mime-type that is requested to be returned is not application/json or the request header or the request body did not have an application/json content-type

Response Examples

Success

Status: 200 OK

```
{
  "id": 5093108584808448,
  "weight": 1,
  "carrier": null,
  "content": "Legos",
  "delivery_date": "1/1/2020",
  "self": "https://elsomr-final.appspot.com/loads/5093108584808448"
}
```

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"
}
```

Failure

Status: 406 Invalid Request

```
{
  "Error": "Request Header not Acceptable"
}
```

Edit a Load – Put

Used to edit one or more attributes, but not necessarily all attributes.

PUT /loads/:load_id

Request

Authorization

None

Request Parameters

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

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
Weight	String	The weight of the load	No
Content	String	The content of the load	No
delivery_date	Integer	The expected date of delivery of the load	No

Request Body Example

```
{
  "weight": 1,
  "delivery_date": "1/1/2020"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	303 OK	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, the load must not be updated, and 400 status code must be returned.
Failure	404 Not Found	No load with this load_id exists
Failure	406 Invalid Request	If the mime-type that is requested to be returned is not application/json or the request header or the request body did not have an application/json content-type

Response Examples

Success

Status: 303 OK

```
{
  "id": 5093108584808448,
  "weight": 1,
  "carrier": null,
  "content": "Legos",
  "delivery_date": "1/1/2020",
  "self": "https://elsomr-final.appspot.com/loads/5093108584808448"
}
```

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"
}
```

Failure

Status: 406 Invalid Request

```
{
  "Error": "Request Header not Acceptable"
}
```

Delete a Load

Allows you to delete a load. If the load being deleted has been assigned to a boat, the boat will also be updated

DELETE /loads/:load_id

Request

Authorization

None

Request Parameters

None

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	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>

Add Load to a Boat

Boat has started shipping a load.

PUT boats/:boat_id/loads/:load_id

Request

Authorization

Required

Request Parameters

None

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Succeeds only if a boat exists with this boat_id, a load exists with this load_id and the load is not already assigned to another boat.
Failure	403 Forbidden	The load is already assigned to another boat
Failure	404 Not Found	No boat with this boat_id exists, and/or no load with this load_id exists.
Failure	401 Invalid Credentials	JWT credentials are missing or invalid
Failure	403 Unauthorized	The boat belongs to someone else

Response Examples

Success

Status: 204 No Content

Failure

Status: 403 Forbidden

```
{
  "Error": "The load is already assigned to another boat."
}
```

Status: 404 Not Found

```
{
```

```
"Error": "The specified boat and/or load don't exist."
}
```

Failure

```
Status: 401 Invalid Credentials
{
  "Error": "invalid or missing credentials"
}
```

Failure

```
Status: 403 Unauthorized

{
  "Error": "The boat belongs to someone else"
}
```

Remove a Load from a Boat

The load has been delivered and unloaded.

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

Request

Authorization

Required

Request Parameters

None

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Succeeds only if a boat exists with this boat_id, a load exists with this load_id and this load is assigned to this boat.
Failure	404 Not Found	No load with this load_id is assigned to the boat at this boat_id.

Failure	401 Invalid Credentials	JWT credentials are missing or invalid
Failure	403 Unauthorized	The boat belongs to someone else

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found
<pre>{ "Error": "No load with this load_id is assigned to a boat with this boat_id" }</pre>

Failure

Status: 401 Invalid Credentials
<pre>{ "Error": "invalid or missing credentials" }</pre>

Failure

Status: 403 Unauthorized
<pre>{ "Error": "The boat belongs to someone else" }</pre>

View all Loads for a Boat

See all the loads the boat is currently shipping

GET /boats/:boat_id/loads

Request

Authorization

None

Request Parameters

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

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No boat with this boat_id exists
Failure	406 Invalid Request	If the mime-type that is requested to be returned is not application/json or the request header or the request body did not have an application/json content-type

Response Examples

Success

Status: 200 OK

```
[
{
  "id": 5099055302574080,
  "content": "Chips",
  "carrier": {
    "id": "4807736529256448",
    "name": "Boat 4",
    "self": "https://elsomr-final.appspot.com/boats/4807736529256448"
  },
  "weight": 25,
  "delivery_date": "4/4/2020",
  "self": "https://elsomr-final.appspot.com/loads/5099055302574080"
}
]
```

Failure

Status: 404 Not Found

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

Failure

Status: 406 Invalid Request

```
{
  "Error": " Request Header not Acceptable"
}
```

View all Boats belonging to a User

View all boats belonging to the current owner.

GET /users/:user_id/boats

Request

Authorization

Required

Request Parameters

Name	Type	Description	Required?
user_id	String	ID of the user (the sub parameter from the JWT token)	Yes

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No user with this user_id exists
Failure	406 Invalid Request	If the mime-type that is requested to be returned is not application/json or the request header or the request body did not have an application/json content-type
Failure	401 Invalid Credentials	JWT credentials are missing or invalid
Failure	403 Unauthorized	The boat belongs to someone else

Response Examples

Success

Status: 200 OK

```
[
  {
    "id": 4887143562346496,
    "name": "Jolly Roger",
    "loads": [],
    "length": 200,
    "type": "Pirate",
    "owner": "114860320530109905851",
    "self": "https://elsomr-final.appspot.com/boats/4887143562346496"
  },
  {
    "id": 5086100271923200,
    "length": 200,
    "type": "Pirate",
```

```
"owner": "114860320530109905851",  
"name": "Jolly Roger",  
"loads": [],  
"self": "https://elsomr-final.appspot.com/boats/5086100271923200"  
}  
]
```

Failure

Status: 404 Not Found

```
{  
  "Error": " No user with this user_id exists"  
}
```

Failure

Status: 406 Invalid Request

```
{  
  "Error": " Request Header not Acceptable"  
}
```

Failure

Status: 401 Invalid Credentials

```
{  
  "Error": "invalid or missing credentials"  
}
```

Failure

Status: 403 Unauthorized

```
{  
  "Error": "The boat belongs to someone else"  
}
```