

API Spec

Fall 2023

Last Update: July, 17 2024

URL: Please see resume for host URL

URL for login/create new user: Please see resume for host URL (click 'login')

Change log	1
Data Model	2
Create a Boat	4
Get a Boat	6
List all Boats	8
Delete a Boat	11
Create a Load	13
Get a Load	14
List all Loads	16
Delete a Load	17
Add a load to a boat.....	18
Remove a load from a boat.....	20
List all loads on a boat.....	22
Edit a Boat (PUT)	24
Edit a Boat (PATCH)	26
Edit a Load (PUT).....	29
Edit a Load (PATCH)	31
Add an Owner	33
List all Owners.....	35

Change log

Version	Change	Date
1.0	Initial version.	Dec 10, 2023
2.0	Host website change	Jul, 17, 2024

Data Model

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

Boats

- Boats is a **non-user entity**.
- Boats has a relationship with the owner entity. “owner” **is a user entity**, therefore endpoints with the root /boats will be protected and require a valid JWT.
- The “owner” property is populated with a unique owner ID when a boat is created (see “Create a boat”). A boat may only have one owner but an owner may have many boats.
- Boats also has a relationship with “loads”. “loads” is a non-user entity. The boats property “loads” is populated with a load entity when it is added to a boat (see “Add a load to a boat”).
- A boat may have many loads but a load may only be carried by one boat.

Property	Data Type	Required?	Notes
id	Integer	Yes	The id of the boat. Datastore automatically generates it.
name	String	Yes	Name of the boat.
type	String	Yes	Type of the boat. E.g., Sailboat, Catamaran, etc.
length	Integer	Yes	The length of the boat in feet.
loads – ‘id’	Integer	No	‘loads’ is a list of load entities being carried by the boat. ‘id’ is a sub-property of a load entity that refers to the id of the load. Datastore automatically generates the id when a load is created. A load is not required. A boat may exist without a load.
self	String	Yes	Contains a link routing to /boats/{{id}} which provides a view of only the boat entity as specified by the ID.
owner	string	Yes	The ‘owner’ property is the owner of the boat. This property is populated by the ‘sub’ property of the jwt when logging in using the UI.

Loads

- Loads is a **non-user entity**.
- Loads has a relationship with “boats” which is also a non-user entity. Loads has no relationship with a user entity and therefore endpoints with the root /loads are not protected and do not require a JWT.
- When a load is added to a boat, the “carrier” property of the load entity is populated with the entity ID of the respective boat.
- A load may have only one carrier (a boat), but a boat may have many loads.

Property	Data Type	Required?	Notes
id	Integer	Yes	The id of the load entity. Datastore automatically generates it.
item	String	Yes	Description of the load.
creation_date	Integer	Yes	Date that the load entity was created
self	String	Yes	Contains a link routing to /loads/{{id}} which provides a view of only the load properties.
volume	Integer	Yes	The volume of the load.
carrier - id	Integer	No	Carrier refers to the boat holding the load. 'id' is a subproperty of 'carrier' and refers to the id of the boat. The id is populated when a load is added to a boat. Carrier is not required. A load may exist without a boat
carrier - self	String	No	Carrier refers to the boat holding the load. 'self' is a subproperty of carrier entity that contains a URL link to all of the boat data. The URL link routes to /boats/{{id}}. Carrier is not required. A load may exist without a boat.

Owner

- Owner is a **user entity**. The unique user ID of an owner corresponds to the “authoID” property of the owner entity. The **authoID is parsed from the ‘sub’ property of a valid JWT** when a user logs in or creates a new account using the login page (see resume)
- When a user logs in for the first time, an owner entity will automatically be created in the datastore, however a returning user with an existing owner entity will not be created a new one.
- **Owner has a relationship with “boats” which is a non-user entity**, therefore endpoints with the root /boats will be protected and require a valid JWT. No other user identifier is required.
- An owner may have many boats but a boat may only have 1 owner.

Property	Data Type	Required?	Notes
id	Integer	Yes	The id owner entity. Datastore automatically generates it.
email	String	Yes	Email of owner
authoID	String	Yes	The unique ID of the owner. The value comes from the ‘sub’ attribute of a valid JWT

name	String	Yes	The name of the owner. This value comes from the 'name' attribute of a valid JWT
self	String	Yes	Contains a link routing to /owner/{{id}} which provides a view of only the owner entity as specified by the ID.

Create a Boat

Allows you to create a new boat using a valid JWT. The 'sub' property is parsed from JWT and used for the "owner" property.

POST /boats

Request

Path Parameters

None

Request Body

Required

Authorization

Required – Bearer Token

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 (e.g. name, type or length), the boat will not be created, and 400 status code will be returned.
Failure	401 Unauthorized	If the request is missing a JWT or the JWT is invalid, the boat will not be created and status code 401 will be returned.
Failure	406 Not Acceptable	If the accept header from the requester does not support JSON, status code 406 will be returned.

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.

Success

```
Status: 201 Created
{
  "id": 123,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "self": "http://127.0.0.1/boats/123",
  "owner": "autho|1234567asdfgh"
  "loads": []
}
```

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: 401 Unauthorized

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

Status: 406 Not Acceptable

```
{
  "Error": "Accept header requester does not support JSON"
}
```

Get a Boat

Allows you to get an existing boat using a valid JWT

```
GET /boats/:boat_id
```

Request Path Parameters

Name	Description
boat_id	ID of the boat

Request Body

None

Authorization

Required - Bearer token

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	401 Unauthorized	If the request is missing a JWT or the JWT is invalid, the boat will not be created and status code 401 will be returned.

Failure	403 Forbidden	If the request contains a valid JWT but the 'sub' property of the JWT does not match the 'authoID' of the boat, status code 403 will be returned.
Failure	406 Not Acceptable	If the accept header from the requester does not support JSON, status code 406 will be returned.

Response Examples

Success

```
Status: 200 OK
{
  "id": 123,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "self": "http://127.0.0.1/boats/123",
  "owner": "autho|1234567asdfgh"
  "loads": []
}
```

Failure

```
Status: 404 Not Found
{
  "Error": "No boat with this boat_id exists"
}
```

```
Status: 401 Unauthorized

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

```
Status: 401 Unauthorized

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

```
Status: 403 Forbidden
{
  "Error": "Requester is not the same as the boat owner"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Accept header requester does not support JSON"
}
```

List all Boats

List all the boats using a valid JWT

GET /boats

Request

Path Parameters

None

Request Body

None

Authorization

Required – Bearer token

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	If the request is missing a JWT or the JWT is invalid, the boat will not be created and status code 401 will be returned.
Failure	406 Not Acceptable	If the accept header from the requester does not support JSON, status code 406 will be returned.

Response Examples

Success

Status: 200 OK

```
{
  "Total Results" = 3,
  "boats": [
    {
      "id": 123,
```



```

    "name": "Sea Witch",
    "type": "Catamaran",
    "length": 28,
    "self": "http://127.0.0.1:8080/1234
    "owner": "autho|jnsdfnkjh21324
    "loads": {[
        "id": 123,
        "volume": 5,
        "item": "LEGO Blocks",
        "creation_date": "10/18/2021",
        "self": "http://127.0.0.1:8080/loads/123",
"carrier": null
    ]}
},
{
    "id": 456,
    "name": "hello stranger",
    "type": "Yacht",
    "length": 45,
    "self": "http://127.0.0.1:8080/1234
    "owner": "autho|jnsdfnkjh21324
    "loads": {[
        "id": 123,
        "volume": 5,
        "item": "LEGO Blocks",
        "creation_date": "10/18/2021",
        "self": "http://127.0.0.1:8080/loads/123",
        "carrier": null
    ]}
}
],
"next": "http://127.0.0.1:8080/boats?limit=5&offset=0"}
}

```

Failure

Status: 401 Unauthorized

```

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

```

Status: 401 Unauthorized

```

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

```

```
}
```

Status: 406 Not Acceptable

```
{
```

"Error": "Accept header requester does not support JSON"

Delete a Boat

Allows you to delete a boat with a valid JWT. Note that if the boat has a load, all 'carrier' attributes of the loads on the boat are updated to null.

DELETE /boats/:boat_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Body

None

Authorization

Required – Bearer token

Response

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 Unauthorized	If the JWT is not valid or is missing, status code 401 will be returned
Failure	403 Forbidden	If the JWT is valid but the boat owner is not the requester, then status code 403 will be returned
Failure	406 Not Acceptable	If the accept header from the requester does not support JSON, status code 406 will be returned.

Response Examples

Success

Status: 204 No Content

Failure

```
Status: 404 Not Found
{
  "Error": "No boat with this boat_id exists"
}
```

```
Status: 401 Unauthorized
{
  "code": "Invalid Header"
  "description": "Invalid header. Use an RS256 signed JWT Access Token"
}
```

```
Status: 401 Unauthorized
{
  "code": "no auth header"
  "description": "Authorization header is missing"
}
```

```
Status: 403 Forbidden
{
  "Error": "Requester is not the same as the boat owner"
}
```

```
Status: 406 Not Acceptable
{
  "Error": "Accept header requester does not support JSON"
}
```

Create a Load

Allows you to create a new load. *Note this is an unprotected endpoint – JWT is not required

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	description or name of the load	Yes
creation_Date	date that the load was created	Yes

Request Body Example

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

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	

Failure	400 Bad Request	If the request is missing the volume, item or creation_date attribute, the load will not be created, and 400 status code will be returned.
Failure	406 Not Acceptable	If the accept header from the requester does not support JSON, status code 406 will be returned.

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.
- Attributes “carrier” and “self” are automatically generated. Carrier is initially set to null and will change once a load is assigned to a boat.

Success

```
Status: 201 Created
{
  "id": 123,
  "volume": 5,
  "item": "LEGO Blocks",
  "creation_date": "10/18/2021",
  "self": "http://127.0.0.1:8080/loads/123",
  "carrier": null
}
```

Failure

```
Status: 400 Bad Request
{
  "Error": "The request object is missing at least one of the required attributes"
}
```

```
Status: 406 Not Acceptable
{
  "Error": "Accept header requester does not support JSON"
}
```

Get a Load

Allows you to get an existing load by ID. *Note, this is an unprotected endpoint - JWT not required

```
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 accept header from the requester does not support JSON, status code 406 will be returned.

Response Examples

Success

```
Status: 200 OK
{
  "id": 134,
  "volume": 10,
  "item": "Cinder Blocks",
  "creation_date": "10/18/2022",
  "self": "http://127.0.0.1:8080/loads/134",
  "carrier": null
}
```

Failure

```
Status: 404 Not Found
{
  "Error": "No load with this load_id exists"
}
```

```
Status: 406 Not Acceptable
{
  "Error": "Accept header requester does not support JSON"
}
```

List all Loads

List all the loads. *Note this is an unprotected endpoint – JWT not required.

GET /loads

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	If the accept header from the requester does not support JSON, status code 406 will be returned.

Response Examples

Success

```
Status: 200 OK
[
  "Total Results": 2,
  "loads": [
    {
      "id": 134,
      "volume": 10,
      "item": "Cinder Blocks",
      "creation_date": "10/18/2022",
      "self": "http://127.0.0.1:8080/loads/134",
      "carrier": {
        "id": "7552"
        "self": "http://127.0.0.1:8080/loads/7552"
      },
    },
    {
      "id": 137,
      "volume": 12,
      "item": "Lego Waffles",
      "creation_date": "10/18/2022",
      "self": "http://127.0.0.1:8080/loads/137",
      "carrier": null
    }
  ]
}
```



```
    }  
  ],  
  "next": "http://127.0.0.1:8080/loads?limit=5&offset=0"  
]
```

Status: 406 Not Acceptable

```
{  
  "Error": "Accept header requester does not support JSON"  
}
```

Delete a Load

Allows you to delete a load. If a load is deleted and associated with a boat, it is also removed from the boat. *Note, this is an unprotected endpoint – JWT not required.

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"

Failure	406 Not Acceptable	If the accept header from the requester does not support JSON, status code 406 will be returned.
---------	--------------------	--

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 404 Not Found
{
  "Error": "No load with this load_id exists"
}
```

```
Status: 406 Not Acceptable
{
  "Error": "Accept header requester does not support JSON"
}
```

Add a load to a boat

Load is placed on a boat. JWT is required.

```
PUT /boats/:boat_id/loads/:load_id
```

Request

Path Parameters

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

Request Body

None

Authorization

Required – Bearer token

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, and a load exists with the load_id.
Failure	403 Forbidden	The load is already loaded on another boat
Failure	404 Not Found	The specified boat and/or load does not exist
Failure	401 Unauthorized	If the request is missing a JWT or the JWT is invalid, the boat will not be created and status code 401 will be returned.
Failure	403 Forbidden	If the request contains a valid JWT but the 'sub' property of the JWT does not match the 'authID' of the boat, status code 403 will be returned.
Failure	406 Not Acceptable	If the accept header from the requester does not support JSON, status code 406 will be returned.

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"
}
```

Status: 401 Unauthorized

```
{
```

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

```
Status: 401 Unauthorized
{
"code": "no auth header"
"description": "Authorization header is missing"
}
```

```
Status: 403 Forbidden
{
"Error": "Requester is not the same as the boat owner"
}
```

```
Status: 406 Not Acceptable
{
"Error": "Accept header requester does not support JSON"
}
```

Remove a load from a boat

A load is taken off of a boat. The number of loads on a boat is decreased by 1. JWT is required

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

Request

Path Parameters

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

Request Body

None

Authorization

Required – Bearer token

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 on the boat.
Failure	404 Not Found	No boat with this boat_id is loaded with the load with this load_id. This could be because no boat with this boat_id exists, or because no load with load_id exists, or even if both boat_id and load_id are valid, the boat with this boat_id does not have the load with this load_id
Failure	401 Unauthorized	If the request is missing a JWT or the JWT is invalid, the boat will not be created and status code 401 will be returned.
Failure	403 Forbidden	If the request contains a valid JWT but the 'sub' property of the JWT does not match the 'authID' of the boat, status code 403 will be returned.
Failure	406 Not Acceptable	If the accept header from the requester does not support JSON, status code 406 will be returned.

Response Examples

Success

```
Status: 204 No Content
```

Failure

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

```
Status: 401 Unauthorized
{
  "code": "Invalid Header"
  "description": "Invalid header. Use an RS256 signed JWT Access Token"
}
```

```
Status: 401 Unauthorized
{
  "code": "no auth header"
  "description": "Authorization header is missing"
}
```

```
Status: 403 Forbidden
{
  "Error": "Requester is not the same as the boat owner"
}
```

```
Status: 406 Not Acceptable
{
  "Error": "Accept header requester does not support JSON"
}
```

List all loads on a boat

Allows you to view all of the loads a boat is carrying.

```
GET /boats/:boat_id/loads
```

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 load with this boat_id exists
Failure	401 Unauthorized	If the request is missing a JWT or the JWT is invalid, the boat will not be created and status code 401 will be returned.

Failure	406 Not Acceptable	If the accept header from the requester does not support JSON, status code 406 will be returned.
---------	--------------------	--

Response Examples

Success

Status: 200 OK

```
"loads": [
  {
    "carrier": {
      "id": 6196,
      "self": "http://127.0.0.1:8080/boats/6196"
    },
    "creation_date": "10/19/2021",
    "id": 4789,
    "item": "Stuffed Animals",
    "self": "http://127.0.0.1:8080/loads/4789",
    "volume": 10
  },
  {
    "carrier": {
      "id": 6196,
```

```
      "self": "http://127.0.0.1:8080/boats/6196"
    },
    "creation_date": "10/20/2021",
    "id": 5352204684754944,
    "item": "Furnitures",
    "self": "http://127.0.0.1:8080/loads/5352",
    "volume": 7
  },
}
```

Failure

Status: 404 Not Found

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

Status: 401 Unauthorized

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

Status: 401 Unauthorized

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

Status: 406 Not Acceptable

```
{
  "Error": "Accept header requester does not support JSON"
}
```

Edit a Boat (PUT)

Allows you to edit a boat with a valid JWT.

PUT /boats/:boat_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Body

Required

Authorization

Required – Bearer token

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	204 No content	
Failure	404 Not Found	No load with this boat_id exists
Failure	401 Unauthorized	If the JWT is not valid or is missing, status code 401 will be returned
Failure	403 Forbidden	If the JWT is valid but the boat owner is not the requester, then status code 403 will be returned
Failure	400 Bad Request	If all 3 attributes are not present in the request body (e.g. name, type or length), then status code 400 will be return.
Failure	406 Not Acceptable	If the accept header from the requester does not support JSON, status code 406 will be returned

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found

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

Status: 401 Unauthorized

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

```
}
```

Status: 401 Unauthorized

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

Status: 403 Forbidden

```
{
  "Error": "Requester is not the same as the boat owner"
}
```

Status: 400 Bad Request

```
{
  "Error": "Incorrect number of attributes or types"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Accept header requester does not support JSON"
}
```

Edit a Boat (PATCH)

Allows you to edit a boat with a valid JWT.

```
PUT /boats/:boat_id
```

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Body

Required

Authorization

Required – Bearer token

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

*At least 1 attribute must be in the response body.

Request Body Examples

```
{
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28
}
```

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

```
{
  "name": "Sea Witch"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No content	
Failure	404 Not Found	No load with this boat_id exists
Failure	401 Unauthorized	If the JWT is not valid or is missing, status code 401 will be returned

Failure	403 Forbidden	If the JWT is valid but the boat owner is not the requester, then status code 403 will be returned
Failure	400 Bad Request	The number of valid attributes must be between 1 and 3, otherwise status code 400 will be returned
Failure	406 Not Acceptable	If the accept header from the requester does not support JSON, status code 406 will be returned

Response Examples

Success

Status: 204 No Content

Failure

```
Status: 404 Not Found
{
  "Error": "No load with this boat_id exists"
}
```

```
Status: 401 Unauthorized
{
  "code": "Invalid Header"
  "description": "Invalid header. Use an RS256 signed JWT Access Token"
}
```

```
Status: 401 Unauthorized
{
  "code": "no auth header"
  "description": "Authorization header is missing"
}
```

```
Status: 403 Forbidden
{
  "Error": "Requester is not the same as the boat owner"
}
```

```
Status: 400 Bad Request
{
  "Error": "Incorrect number of attributes or types"
}
```

```
Status: 406 Not Acceptable
{
  "Error": "Accept header requester does not support JSON"
```

```
}
```

Edit a Load (PUT)

Allows you to edit a load. *Note this is an unprotected endpoint – JWT not required.

```
PUT /loads/:load_id
```

Request

Path Parameters

Name	Description
load_id	ID of the load

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
volume	number of items	Yes
item	Description of the item. E.g., Widgets, cars, lumber etc.	Yes
creation_date	The date the load was created	Yes

Request Body Example

```
{
  "volume": "5000",
  "item": "Lumber",
  "creation_date": "10/18/2023"
}
```

Response

Response Body Format

JSON

Response Statuses

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

Success	204 No content	
Failure	404 Not Found	No load with this boat_id exists
Failure	400 Bad Request	If all 3 attributes are not present in the request body (e.g. name, type or length), then status code 400 will be return.
Failure	406 Not Acceptable	If the accept header from the requester does not support JSON, status code 406 will be returned

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 404 Not Found
{
  "Error": "No load with this boat_id exists"
}
```

```
Status: 400 Bad Request
{
  "Error": "Incorrect number of valid attributes"
}
```

```
Status: 406 Not Acceptable
{
  "Error": "Accept header requester does not support JSON"
}
```

Edit a Load (PATCH)

Allows you to edit a boat with a valid JWT.

```
PUT /loads/:load_id
```

Request

Path Parameters

Name	Description
load_id	ID of the load

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
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

*At least 1 attribute must be in the response body.

Request Body Examples

```
{
  "volume": 250,
  "item": "lamborghini",
  "creation_date": "12/25/2022"
}
```

```
{
  "volume": 100,
  "type": "911 gt3 RS",
}
```

```
{
```

```
"volume": 57
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No content	
Failure	404 Not Found	No load with this load_id exists
Failure	400 Bad Request	The number of valid attributes must be between 1 and 3, otherwise status code 400 will be returned
Failure	406 Not Acceptable	If the accept header from the requester does not support JSON, status code 406 will be returned

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found

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

Status: 400 Bad Request

```
{
  "Error": "Incorrect number of attributes or types"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Accept header requester does not support JSON"
}
```


Add an Owner

Allows you to add an owner. ***NOTE:** this endpoint is automatically called by the UI when a user logs in or a new user is created. If a user already exists, an entity will not be created. If a user creates a new account, a new entity will be created. The entity status creation status will be listed at the bottom of the UI. The JWT verification is handled by the Autho login.

POST /owner

Request

Path

Parameters

none Request

Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
authoID	The value of the 'sub' property in a JWT	Yes
name	The value of the 'name' property in a JWT	Yes
email	The value of the 'email' property in a JWT	Yes

Request Body Example

```
{
  "authoID": "autho|1234543fsfsdr3",
  "name": "bob@gmail.com",
  "email": "bob@gmail.com"
}
```

*Request body must contain all 3 attributes.

Response

Response Body Format

JSON

Response Statuses

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

Success	204 No content	If a user entity already exists, status code 204 will be returned
Success	201 Created	If a user entity does not exist, status code 201 will be returned
Failure	400 Bad Request	If all 3 attributes are not present in the request body (e.g. name, email or Autho), then status code 400 will be returned.
Failure	406 Not Acceptable	If the accept header from the requester does not support JSON, status code 406 will be returned

Response Examples

Success

Status: 204 No Content

```
{
  "Note": "Entity with AuthoID already exists in the datastore. New user id NOT created"
}
```

Status: 201 Created

```
{
  "Note": "New user created in the datastore"
}
```

Failure

Status: 400 Bad Request

```
{
  "Error": "Request body does not contain all valid elements"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Accept header requester does not support JSON"
}
```

List all Owners

Allows you see the list of all owners

GET /owner

Request

Path

Parameters

none Request

Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Returns the list of owners
Failure	406 Not acceptable	If the accept header from the requester does not support JSON, status code 406 will be returned

Response Examples

Success

```
Status: 200 OK
[
  {
    "authoID": "auth0|oiu20394uujf1kns",
    "email": "draymondGreen@gmail.com",
    "id": 98745683032,
    "name": "draymondGreen@gmail.com",
    "self": "http://127.0.0.1:8080/owner/98745683032"
  },
  {
    "authoID": "auth0|88u45hnfojweru4",
    "email": "compsci@gmail.com ",
    "id": 234087759874,
    "name": "compsci@gmail.com",
    "self": "http://127.0.0.1:8080/owner/234087759874"
  },
]
```

```
{
  "authoID": "auth0|8038475hngnkjnurnbw",
  "email": "stephCurry@gmail.com",
  "id": 1230985930059,
  "name": "stephCurry@gmail.com",
  "self": "http://127.0.0.1:8080/owner/1230985930059"
}
```

Failure

```
Status: 406 Not Acceptable
{
  "Error": "Accept header requester does not support JSON"
}
```