

Portfolio Project

Breanna Moore

ONID: moorbrea

CS 493: Cloud Application Development

Fall 2021

Oregon State University

Last Update: Dec 2, 2021

App URL.....	1
Account creation/login URL	2
Change log.....	2
Data Model	3
Get all Registered Users.....	4
Create a Boat	5
Get a Boat	7
Get all Boats for a specified User	9
Edit a Boat with PATCH	11
Edit a Boat with PUT	14
Delete a Boat.....	17
Create a Load	18
Get a Load	20
List all Loads	21
Edit a Load with PATCH.....	23
Edit a Load with PUT	26
Delete a Load	28
Assign Load to a Boat.....	29
A Load is Removed from a Boat.....	31
Get all Loads Assigned to a Boat.....	32

App URL

<https://final-moorbrea.wl.r.appspot.com>

Account creation/login URL

Note: Click the “Log In” button at the bottom of the page to login.

<https://final-moorbrea.wl.r.appspot.com>

Change log

Version	Change	Date
2.0	Finished routes for boats and loads	Nov, 30, 2021
1.0	Initial version.	Nov 23, 2021

Data Model

The app stores two kinds of entities in Datastore, Boats and Loads. Loads can be assigned to Boats and each Boat can carry multiple loads. Each load can only be assigned to one Boat. Users can create an account or login through Auth0's authentication service. A Boat can only be created, updated, or deleted with a valid JWT supplied in the request's Authorization header. The Boat property owner is the user ID encoded in the valid JWT. The owner of a specified boat is the only user that can modify or view the details of that boat.

Boats

Property	Data Type	Notes
id	Integer	The id of the boat. Datastore automatically generates it. Don't add it yourself as a property of the entity.
name	String	Name of the boat. No special characters allowed. String must be no more than 30 characters long.
type	String	Type of the boat. E.g., Sailboat, Catamaran, etc. No special characters allowed. String must be no more than 30 characters long.
length	Integer	The length of the boat in feet. Length must be a number greater than zero.
Owner	String	The authorized user's Auth0 ID. This value is pulled from the JWT in the Authorization request header.
Loads	Array	If the boat has been assigned any loads, each load's id will be stored as an object in the array. A boat can hold multiple loads. Otherwise, loads is null.

Loads

Property	Data Type	Notes
id	Integer	The id of the load. Datastore automatically generates it. Don't add it yourself as a property of the entity.
volume	Integer	The volume of the load. Volume must be a number greater than zero.
carrier	array	If the load has been assigned to a boat, the boat's id will be stored as an object in the carrier array. A load can only be assigned to one carrier at a time. Otherwise, carrier is null.
content	String	The description of what the load contains. Content must be at most 75 characters in length.
creation_date	String	The date that the load was created. The date must be a formatted as MM/DD/YYYY.

Users

Property	Data Type	Notes
email	String	The email address of a registered user.
user_id	Integer	The ID of a registered user. This value is automatically

Get all Registered Users

Lists the user ID and email of all registered users.

GET /users

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	Invalid Token. Error message will be sent by Auth0's Management API.

Success

Status: 200 OK

```
[
  {
    "email": "tobi@thedog.com",
    "user_id": "auth0|619f00a5af07b300718a54ce"
  },
  {
    "email": "ludo@kingcat.com",
    "user_id": "auth0|61a17e10c49d580069259631"
  }
]
```

Failure

Status: 401 Unauthorized

Invalid Token

Create a Boat

Allows a user with a valid JWT to create a new boat.

POST /boats

Request

Path Parameters

None

Request Body

Required

Request Header Authorization

Required

Request Header Authorization Example

```
{
  "Authorization": "Bearer <valid JWT>"
}
```

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	400 status code will be sent for the following scenarios: The request is missing any of the 3 required attributes or contains extraneous attributes. The data type of name, type, or length are incorrect

		<p>The attributes name and type contain special characters or are more than 30 characters in length.</p> <p>The attribute length is not greater than zero.</p>
Failure	401 Unauthorized	Missing or invalid JWT
Failure	403 Forbidden	There already is a boat with the specified boat name.
Failure	406 Not Acceptable	The client wants an unsupported MIME type in the response.
Failure	415 Unsupported Media Type	The client sends a request in an unsupported MIME type.

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. Your app must send back this value in the response body as shown in the example.
- The boat property owner will be pulled from the sub object value stored in the valid, encoded JWT.
- The `self` attribute will contain the live link to the REST resource corresponding to this boat. In other words, this is the URL to get this newly created boat.

Success

Status: 201 Created

```
{
  "id": 123,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "owner": "auth0|619f00a5af07b300718a54ce",
  "loads": null,
  "self": "https://<my-app>/boats/123"
}
```

Failure

Status: 400 Bad Request

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

Status: 400 Bad Request

```
{
  "Error": "The request object contains invalid number of attributes"
}
```

Status: 400 Bad Request

```
{
  "Error": "The attribute name must be a string of at most 30 characters"
}
```

Status: 400 Bad Request

<pre>{ "Error": "The attribute name must not contain special characters" }</pre>
Status: 400 Bad Request <pre>{ "Error": "The attribute length must be a number greater than zero" }</pre>
Status: 401 Unauthorized <pre>{ "Error": "Missing or invalid JWT" }</pre>
Status: 403 Forbidden <pre>{ "Error": "A boat with the specified name already exists" }</pre>
Status: 406 Not Acceptable Not Acceptable
Status: 415 Unsupported Media Type Server only accepts application/json data.

Get a Boat

Allows you to get an existing boat with a valid JWT

GET /boats/:boat_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Header Authorization

Required

Request Header Authorization Example

<pre>{ "Authorization": "Bearer <valid JWT>" }</pre>
--

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	Missing or Invalid JWT
Failure	404 Not Found	No boat with this boat_id exists
Failure	406 Not Acceptable	The client wants an unsupported MIME type in the response.
Failure	500 Internal Server Error	Server Error

Response Examples

Success

Status: 200 OK
<pre>{ "id": 123, "name": "Sea Witch", "type": "Catamaran", "length": 28, "owner": "auth0 619f00a5af07b300718a54ce", "loads": null, "self": "https://<my-app>/boats/123" }</pre>

Failure

Status: 401 Unauthorized
<pre>{ "Error": "Missing or invalid JWT" }</pre>
Status: 404 Not Found
<pre>{ "Error": "No boat with this boat_id exists" }</pre>
Status: 406 Not Acceptable
Not Acceptable
Status: 500 Internal Server Error
Server Error

Get all Boats for a specified User

Allows you to get all boats for a user with a valid JWT

GET /boats

Request

Path Parameters

None

Request Body

None

Request Header Authorization

Required

Request Header Authorization Example

```
{
  "Authorization": "Bearer <valid JWT>"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	Invalid Token. Error message will be sent by Auth0's Management API.
Failure	406 Not Acceptable	The client wants an unsupported MIME type in the response.
Failure	415 Unsupported Media Type	The client sends a request in an unsupported MIME type.

Response Examples

Success

Status: 200 OK

```
{
  "items":
  [
    {
      "id": "5126591025774592",
      "name": "Gone Fishin",
      "type": "Pontoon",
    }
  ]
}
```

```

    "length": 40,
    "owner": "auth0|619f00a5af07b300718a54ce",
    "loads": null,
    "self": https://final-moorbrea.wl.r.appspot.com/boats/5126591025774592
  },
  {
    "id": "5142858919051264",
    "name": "Mega Boat",
    "type": "Yacht",
    "length": 200,
    "owner": "auth0|619f00a5af07b300718a54ce",
    "loads": null,
    "self": https://final-moorbrea.wl.r.appspot.com/boats/5142858919051264
  },
  {
    "id": "5192424284487680",
    "name": "Voyager",
    "type": "Yacht",
    "length": 63,
    "owner": "auth0|619f00a5af07b300718a54ce",
    "loads": null,
    "self": https://final-moorbrea.wl.r.appspot.com/boats/5192424284487680
  },
  {
    "id": "5705808872472576",
    "name": "Sea Witch",
    "type": "Catamaran",
    "length": 28,
    "owner": "auth0|619f00a5af07b300718a54ce",
    "loads": [
      {
        "id": 5087502092206080,
        "self": https://final-moorbrea.wl.r.appspot.com/loads/5087502092206080
      },
      {
        "id": 5138716758638592,
        "self": https://final-moorbrea.wl.r.appspot.com/loads/5138716758638592
      },
      {
        "id": 6213401999048704,
        "self": https://final-moorbrea.wl.r.appspot.com/loads/6213401999048704
      }
    ],
    "self": https://final-moorbrea.wl.r.appspot.com/boats/5705808872472576
  },
  {
    "id": "5712837116690432",
    "name": "Boaty",

```

```

"type": "Cruise Ship",
"length": 300,
"owner": "auth0|619f00a5af07b300718a54ce",
"loads": null,
"self": https://final-moorbrea.wl.r.appspot.com/boats/5712837116690432
}
],
"count":8,
"next": https://final-moorbrea.wl.r.appspot.com?cursor=CiwSJmoQbX5maW5hbC1tb29yYnJlYXISCxIFQm9hdHMYgICA2Lj5kgO MGAAgAA==
}

```

Failure

Status: 401 Unauthorized
<pre>{ "Error": "Missing or invalid JWT" }</pre>
Status: 406 Not Acceptable
Not Acceptable
Status: 415 Unsupported Media Type
Server only accepts application/json data.

Edit a Boat with PATCH

Allows user to edit any subset of attributes for a boat except for the boat's ID, loads, and owner.

PATCH /boats/:boat_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Header Authorization

Required

Request Header Authorization Example

```

{
  "Authorization": "Bearer <valid JWT>"
}

```

Request Body

At least one attribute key/value pair is required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	The name of the boat.	Optional
type	The type of the boat. E.g., Sailboat, Catamaran, etc.	Optional
length	Length of the boat in feet.	Optional

Request Body Example

```
{
  "name": "Serpent",
  "length": 50
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	400 status code will be sent for the following scenarios: The request contains no, null, or contains extraneous attributes. At least one attribute needs to be in the request body. The data type of name, type, or length are incorrect The attributes name and type contain special characters or are more than 30 characters in length.
Failure	401 Unauthorized	Missing or invalid JWT
Failure	404 Not Found	No boat with this boat_id exists
Failure	403 Forbidden	The attribute name has already been assigned to a different boat in the datastore.
Failure	405 Method Not Allowed	Client didn't include the boat_id in the request URL.
Failure	406 Not Acceptable	The client wants an unsupported MIME type in the response.
Failure	415 Unsupported Media Type	The client sends a request in an unsupported MIME type.

Response Examples

- The `self` attribute will contain the live link to the REST resource corresponding to this boat.

Success

```
Status: 200 OK
```

```
{
  "id": 123,
  "name": "Serpent",
  "type": "Catamaran",
  "length": 50,
  "owner": "auth0|619f00a5af07b300718a54ce",
  "loads": null,
  "self": "https://<my-app>/boats/123"
}
```

Failure

Status: 400 Bad Request

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

Status: 400 Bad Request

```
{
  "Error": "The request object contains invalid number of attributes"
}
```

Status: 400 Bad Request

```
{
  "Error": "The attribute name must be a string of at most 30 characters"
}
```

Status: 400 Bad Request

```
{
  "Error": "The attribute name must not contain special characters"
}
```

Status: 400 Bad Request

```
{
  "Error": "The attribute length must be a number greater than zero"
}
```

Status: 401 Unauthorized

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

Status: 403 Forbidden

```
{
  "Error": "A boat with the specified name already exists"
}
```

Status: 404 Not Found

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

}
Status: 405 Method Not Allowed
Response Header
<pre>{ "Accept": "GET, POST" }</pre>
Status: 406 Not Acceptable
Not Acceptable
Status: 415 Unsupported Media Type
Server only accepts application/json data.

Edit a Boat with PUT

Allows user to update all attributes for a boat except for the boat's ID, owner, and loads.

PUT /boats/:boat_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Header Authorization

Required

Request Header Authorization Example

<pre>{ "Authorization": "Bearer <valid JWT>" }</pre>
--

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
name	The name of the boat.	required
type	The type of the boat. E.g., Sailboat, Catamaran, etc.	required
length	Length of the boat in feet.	required

Request Body Example

{

```
"name": "Serpent",  
"type": "Sailboat",  
"length": 50  
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	400 status code will be sent for the following scenarios: The request is missing any of the 3 required attributes or contains extraneous attributes. The data type of name, type, or length are incorrect The attributes name and type contain special characters or are more than 30 characters in length. The attribute length is less than or equal to zero.
Failure	401 Unauthorized	Missing or invalid JWT
Failure	404 Not Found	No boat with this boat_id exists
Failure	403 Forbidden	The attribute name has already been assigned to a different boat in the datastore.
Failure	405 Method Not Allowed	Client didn't include the boat_id in the request URL.
Failure	406 Not Acceptable	The client wants an unsupported MIME type in the response.
Failure	415 Unsupported Media Type	The client sends a request in an unsupported MIME type.

Response Examples

- The `self` attribute will contain the live link to the REST resource corresponding to this boat.

Success

Status: 200 OK

```
{  
  "id": 123,  
  "name": "Serpent",  
  "type": "Sailboat",  
  "length": 50,  
  "owner": "auth0|619f00a5af07b300718a54ce",  
  "loads": null,  
  "self": "https://<my-app>/boats/123"  
}
```

Failure

Status: 400 Bad Request <pre>{ "Error": "The request object is missing at least one of the required attributes" }</pre>
Status: 400 Bad Request <pre>{ "Error": "The request object contains invalid number of attributes" }</pre>
Status: 400 Bad Request <pre>{ "Error": "The attribute name must be a string of at most 30 characters" }</pre>
Status: 400 Bad Request <pre>{ "Error": "The attribute name must not contain special characters" }</pre>
Status: 401 Unauthorized <pre>{ "Error": "Missing or invalid JWT" }</pre>
Status: 403 Forbidden <pre>{ "Error": "A boat with the specified name already exists" }</pre>
Status: 404 Not Found <pre>{ "Error": "No boat with this boat_id exists" }</pre>
Status: 405 Method Not Allowed Response Header <pre>{ "Accept": "GET, POST" }</pre>
Status: 406 Not Acceptable Not Acceptable
Status: 415 Unsupported Media Type

Server only accepts application/json data.

Delete a Boat

Allows a user to delete a boat.

DELETE /boats/:boat_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Header Authorization

Required

Request Header Authorization Example

```
{
  "Authorization": "Bearer <valid JWT>"
}
```

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	Missing or invalid JWT
Failure	404 Not Found	No boat with this boat_id exists
Failure	405 Method Not Allowed	Client didn't include the boat_id in the request URL.

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Unauthorized

<pre>{ "Error": "Missing or invalid JWT" }</pre>
Status: 404 Not Found
<pre>{ "Error": "No boat with this boat_id exists" }</pre>
Status: 405 Method Not Allowed
Response Header
<pre>{ "Accept": "GET, POST" }</pre>

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
content	The description of the load's content.	Yes
creation_date	The date (MM/DD/YYYY) the load was created.	Yes

Request Body Example

<pre>{ "volume": 5, "type": "LEGO Blocks", "creation_date": "10/18/2021" }</pre>
--

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	400 status code will be sent for the following scenarios: The request is missing any of the 3 required attributes or contains extraneous attributes. The data type of volume, content, or creation_date are incorrect. The attribute content is more than 75 characters in length. The attribute creation_date is not in the format MM/DD/YYYY The attribute volume is not greater than zero.
Failure	406 Not Acceptable	The client wants an unsupported MIME type in the response.
Failure	415 Unsupported Media Type	The client sends a request in an unsupported MIME type.

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.
- The value of the attribute `carrier` is the ID of the boat currently carrying the load. If there is no boat assigned the load, the value of `carrier` is null.
- The value of the attribute `self` is a live link to the REST resource corresponding to this load.
- Loads are public and don't require the request to have an Authorization header.

Success

Status: 201 Created
<pre>{ "id": 567 "volume": 5, "content": "LEGO Blocks", "creation_date": "10/18/2021", "carrier": null, "self": "https://<my-app>/loads/567"</pre>

```
}
```

Failure

Status: 400 Bad Request

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

Status: 400 Bad Request

```
{
  "Error": "The request object contains invalid number of attributes"
}
```

Status: 400 Bad Request

```
{
  "Error": "The attribute content must be a string of at most 75 characters"
}
```

Status: 400 Bad Request

```
{
  "Error": "The attribute creation_date must be in the format MM/DD/YYYY"
}
```

Status: 400 Bad Request

```
{
  "Error": "The attribute volume must be a number greater than zero"
}
```

Status: 406 Not Acceptable

Not Acceptable

Status: 415 Unsupported Media Type

Server only accepts application/json data.

Get 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 client wants an unsupported MIME type in the response.
Failure	415 Unsupported Media Type	The client sends a request in an unsupported MIME type.

Response Examples

Success

Status: 200 OK

```
{
  "id": 567
  "volume": 5,
  "content": "LEGO Blocks",
  "creation_date": "10/18/2021",
  "carrier": null,
  "self": "https://<my-app>/loads/567"
}
```

Failure

Status: 404 Not Found

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

Status: 406 Not Acceptable

Not Acceptable

Status: 415 Unsupported Media Type

Server only accepts application/json data.

List all Loads

List all the loads.

GET /loads

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	The client wants an unsupported MIME type in the response.
Failure	415 Unsupported Media Type	The client sends a request in an unsupported MIME type.

Response Examples

Success

Status: 200 OK

```
{
  "items": [
    {
      "id": "4806027115495424",
      "volume": 40,
      "content": "Nintendo Pro controllers",
      "creation_date": "09/20/2021",
      "carrier": [{
        "id": 5705808872472576
      }],
      "self": https://final-moorbrea.wl.r.appspot.com/loads/4806027115495424
    },
    {
      "id": "5087502092206080",
      "volume": 5,
      "content": "LEGO Blocks",
      "creation_date": "10/18/2021",
      "carrier": [{
        "id": 5705808872472576
      }],
      "self": https://final-moorbrea.wl.r.appspot.com/loads/5087502092206080
    },
    {
      "id": "5138716758638592",
      "volume": 19,
      "content": "Boxes of milk chocolate bars",
      "creation_date": "10/22/2021",
      "carrier": [{
        "id": 5705808872472576
      }],
      "self": https://final-moorbrea.wl.r.appspot.com/loads/5138716758638592
    }
  ]
}
```

```

  }},
  "self": https://final-moorbrea.wl.r.appspot.com/loads/5138716758638592
},
{
  "id": "5368977068916736",
  "volume": 17,
  "content": "Blue couches",
  "creation_date": "11/14/2021",
  "carrier": null,
  "self": https://final-moorbrea.wl.r.appspot.com/loads/5368977068916736
},
{
  "id": "5685839489138688",
  "volume": 43,
  "content": "Boxes of instant ramen",
  "creation_date": "09/06/2021",
  "carrier": null,
  "self": https://final-moorbrea.wl.r.appspot.com/loads/5685839489138688
}},
"count": 8,
"next": https://final-moorbrea.wl.r.appspot.com?cursor=CiwSJmoQbX5maW5hbC1tb29yYnJlYXISCxIFTG9hZHM5gICA2NrnjAoMGAAGAA==
}

```

Failure

Status: 406 Not Acceptable
Not Acceptable
Status: 415 Unsupported Media Type
Server only accepts application/json data.

Edit a Load with PATCH

Allows user to edit any subset of attributes for a load except for the load's ID and carrier.

PATCH /loads/:load_id

Request

Path Parameters

Name	Description
load_id	ID of the load

Request Body

At least one attribute key/value pair is required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
volume	The volume of the load.	Optional
content	The description of the load's content.	Optional
creation_date	The date (MM/DD/YYYY) the load was created.	Optional

Request Body Example

```
{
  "volume": 10
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	400 status code will be sent for the following scenarios: The request contains no, null, or contains extraneous attributes. At least one attribute needs to be in the request body. The data type of volume, content, or creation_date are incorrect The attribute content is more than 75 characters in length. The attribute creation_date is not in the format MM/DD/YYYY The attribute volume is not a number greater than zero.
Failure	404 Not Found	No load with this load_id exists
Failure	405 Method Not Allowed	Client didn't include the load_id in the request URL.
Failure	406 Not Acceptable	The client wants an unsupported MIME type in the response.
Failure	415 Unsupported Media Type	The client sends a request in an unsupported MIME type.

Response Examples

- The `self` attribute will contain the live link to the REST resource corresponding to this load.

Success

Status: 200 OK

```
{
  "id": 567
  "volume": 10,
  "content": "LEGO Blocks",
  "creation_date": "10/18/2021",
  "carrier": null,
  "self": "https://<my-app>/loads/567"
}
```

Failure

Status: 400 Bad Request

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

Status: 400 Bad Request

```
{
  "Error": "The request object contains invalid number of attributes"
}
```

Status: 400 Bad Request

```
{
  "Error": "The attribute content must be a string of at most 75 characters"
}
```

Status: 400 Bad Request

```
{
  "Error": "The attribute creation_date must be in the format MM/DD/YYYY"
}
```

Status: 400 Bad Request

```
{
  "Error": "The attribute volume must be a number greater than zero"
}
```

Status: 404 Not Found

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

Status: 405 Method Not Allowed

Response Header

```
{
  "Accept": "GET, POST"
}
```

Status: 406 Not Acceptable

Not Acceptable

Status: 415 Unsupported Media Type

Server only accepts application/json data.

Edit a Load with PUT

Allows user to edit attributes for a load except for the load's ID and carrier.

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	The volume of the load.	Yes
content	The description of the load's content.	Yes
creation_date	The date (MM/DD/YYYY) the load was created.	Yes

Request Body Example

```
{
  "volume": 15,
  "content": "Nanoblocks",
  "creation_date": "11/28/2021"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	400 status code will be sent for the following scenarios: The request contains no, null, or contains extraneous attributes.

		<p>The data type of volume, content, or creation_date are incorrect.</p> <p>The attribute content is more than 75 characters in length.</p> <p>The attribute creation_date is not in the format MM/DD/YYYY</p> <p>The attribute volume is not a number greater than zero.</p>
Failure	404 Not Found	No load with this load_id exists
Failure	405 Method Not Allowed	Client didn't include the load_id in the request URL.
Failure	406 Not Acceptable	The client wants an unsupported MIME type in the response.
Failure	415 Unsupported Media Type	The client sends a request in an unsupported MIME type.

Response Examples

- The `self` attribute will contain the live link to the REST resource corresponding to this load.

Success

Status: 200 OK

```
{
  "id": 567
  "volume": 15,
  "content": "Nanoblocks",
  "creation_date": "11/28/2021",
  "carrier": null,
  "self": "https://<my-app>/loads/567"
}
```

Failure

Status: 400 Bad Request

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

Status: 400 Bad Request

```
{
  "Error": "The request object contains invalid number of attributes"
}
```

Status: 400 Bad Request

```
{
  "Error": "The attribute content must be a string of at most 75 characters"
}
```

Status: 400 Bad Request <pre>{ "Error": "The attribute creation_date must be in the format MM/DD/YYYY" }</pre>
Status: 400 Bad Request <pre>{ "Error": "The attribute volume must be a number greater than zero" }</pre>
Status: 404 Not Found <pre>{ "Error": "No load with this load_id exists" }</pre>
Status: 405 Method Not Allowed Response Header <pre>{ "Accept": "GET, POST" }</pre>
Status: 406 Not Acceptable Not Acceptable
Status: 415 Unsupported Media Type Server only accepts application/json data.

Delete a Load

Allows a user to delete a load.

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	405 Method Not Allowed	Client didn't include the load_id in the request URL.

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found

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

Status: 405 Method Not Allowed

Response Header

```
{  
  "Accept": "GET, POST"  
}
```

Assign Load to a Boat

Allows user with valid JWT to assign a load to their 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 Header Authorization

Required

Request Header Authorization Example

```
{  
  "Authorization": "Bearer <valid JWT>"  
}
```

```
}
```

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Succeeds only if a boat exists with this boat_id, a load exists with this load_id and the load has not been assigned to a boat. Valid JWT that matches owner of the boat is also required.
Failure	401 Unauthorized	Missing or invalid JWT.
Failure	403 Forbidden	There is already a boat assigned to the load.
Failure	404 Not Found	The specified boat and/or load does not exist

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Unauthorized

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

Status: 403 Forbidden

```
{  
  "Error": "The load has already been assigned to a boat"  
}
```

Status: 404 Not Found

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

Comment

A load cannot be assigned to multiple boats.

A Load is Removed from a Boat

Allows a user with a valid JWT to remove a load from their 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 Header Authorization

Required

Request Header Authorization Example

```
{
  "Authorization": "Bearer <valid JWT>"
}
```

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Succeeds only if a boat exists with this boat_id, a load exists with this load_id and this load has been assigned to this boat. The boat's owner must also match the valid JWT.
Failure	401 Unauthorized	Missing or invalid JWT
Failure	404 Not Found	No load with this load_id and/or boat with this boat_id exists.
Failure	403 Forbidden	The load is not assigned to the boat with this boat_id.

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Unauthorized

```
{
```

"Error": "Missing or invalid JWT"
<pre>{ "Error": "The load is not assigned to the specified boat" }</pre>
<pre>Status: 404 Not Found { "Error": "The specified boat and/or load does not exist" }</pre>

Get all Loads Assigned to a Boat

Allows a user to view a list of all the loads assigned to a specified boat. This endpoint does not require authorization.

GET /boats/:boat_id/loads

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Succeeds only if a boat exists with this boat_id and there are loads assigned to this boat.
Failure	404 Not Found	No boat with this boat_id exists or no loads have been assigned to this boat.
Failure	406 Not Acceptable	The client wants an unsupported MIME type in the response.
Failure	415 Unsupported Media Type	The client sends a request in an unsupported MIME type.

Response Examples

Success

Status: 200 OK

```
[
{
  "carrier": [{
    "id": 5705808872472576,
    "name": "Sea Witch",
    "self": https://final-moorbrea.wl.r.appspot.com/boats/5705808872472576
  }],
  "creation_date": "09/20/2021",
  "volume": 40,
  "content": "Nintendo Pro controllers",
  "id": "4806027115495424",
  "self": https://final-moorbrea.wl.r.appspot.com/loads/4806027115495424
},
{
  "carrier": [{
    "id": 5705808872472576,
    "name": "Sea Witch",
    "self": https://final-moorbrea.wl.r.appspot.com/boats/5705808872472576
  }],
  "content": "LEGO Blocks",
  "creation_date": "10/18/2021",
  "volume": 5,
  "id": "5087502092206080",
  "self": https://final-moorbrea.wl.r.appspot.com/loads/5087502092206080
},
{
  "volume": 19,
  "creation_date": "10/22/2021",
  "content": "Boxes of milk chocolate bars",
  "carrier": [{
    "id": 5705808872472576,
    "name": "Sea Witch",
    "self": https://final-moorbrea.wl.r.appspot.com/boats/5705808872472576
  }],
  "id": "5138716758638592",
  "self": https://final-moorbrea.wl.r.appspot.com/loads/5138716758638592
},
{
  "content": "Nintendo Switch consoles",
  "carrier": [{
    "id": 5705808872472576,
    "name": "Sea Witch",
    "self": https://final-moorbrea.wl.r.appspot.com/boats/5705808872472576
  }],
  "volume": 7,
  "creation_date": "09/02/2021",
  "id": "6213401999048704",
  "self": https://final-moorbrea.wl.r.appspot.com/loads/6213401999048704
}
]
```

```
}  
]
```

Failure

Status: 404 Not Found

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

Status: 406 Not Acceptable

Not Acceptable

Status: 415 Unsupported Media Type

Server only accepts application/json data.