

Final Project

CS 493: Cloud Application Development

Spring 2020

Oregon State University

Last Update: June 8, 2020

URL	1
Data Model	2
Login/Create Owner Account	4
Get Owners	5
Create a Boat	6
Get a Boat	8
List all Boats	10
Edit a Boat (PUT)	13
Edit a Boat (PATCH).....	15
Delete a Boat.....	18
Create a Load	20
Get a Load	22
List all Loads	24
Edit a Load (PUT).....	26
Edit a Load (PATCH)	28
Delete a Load	30
Load Gets Added to Boat	31
Load Gets Removed From Boat	33

URL

[https:// cs-493-final-279202.wl.r.appspot.com](https://cs-493-final-279202.wl.r.appspot.com)

Data Model

Entities

User

Property	Type	Notes
uniqueID	String	This is auto generated with each account creation.

Example

```
{
  "uniqueID": " 108240556924460704143",
  "id": "5636645067948032"
}
```

Comment

- The uniqueID is generated from the sub value from a valid JWT token.
- For request that requires a user identifier, the person making the request needs to supply a valid JWT token as Bearer token in authorization header as a part of the request. The backend server then validates the token and retrieves the sub value in the token to map it to a registered user via user's uniqueID.
 - Examples of requests that require a user identifier includes all CRUD action to boats since Boat entity is tied to a user.

Boat

Property	Type	Notes
Name	String	Required and provided by the user.
Type	String	Required and provided by the user.
Length	Number	Required and provided by the user.
Owner	String	Not required. The field will be populated by the server with each successful authenticated requests. The field is the uniqueID value of the respective User.
Loads	Object Array	Not required. The field will be populated by the server with each successful authenticated requests.

Example

```
{
  {
    "id": "5141184150241280",
    "name": "Odyssey",
    "type": "Yatch",
    "length": 99,
    "owner": " 108240556924460704143",
    "loads": [
      {
        "id": "5642061785071616",
        "self": "http://<your-app>/loads/5642061785071616"
      },
      {
        "id": "6275787070111744",
        "self": "http://<your-app>/loads/6275787070111744"
      }
    ]
  }
}
```

```

    ],
    "self": "http://<your-app>/boats/5141184150241280"
  }
}

```

Load

Property	Type	Notes
Weight	Number	Required and provided by the user.
Content	String	Required and provided by the user.
Delivery Date	Date	Required and provided by the user.
Carrier	Object Array	Not required. The field will be populated by the server with each successful requests.

Example

```

{
  {
    "carrier": [{
      "boat_id": "4730785932247040",
      "self": http://<your-app>/boats/4730785932247040
    }],
    "content": "Down",
    "weight": 1,
    "delivery_date": "1/1/2020",
    "id": "5285439082594304",
    "self": "http://<your-app>/loads/5285439082594304"
  }
}

```

Comment

- Boat and Load are non-user entities. Load entity can be added to a boat. If a boat has load, the boat's id will be displayed under the "Carrier" property of the load; the load's id will be displayed under "Loads" property of the boat.
- A boat can have more than one loads but a load can only be on one boat.
- All boats are associated with a user. Ownership cannot be transferred.
- Deletion of a load will remove the load from the boat.
- Deletion of a loaded boat will clear the "Carrier" property of the load. The load, however, will not be deleted.

Login/Create Owner Account

Allows you to authorize my web application to retrieve your account information and register new owner accounts

https://cs-493-final-279202.wl.r.appspot.com

Note

Please visit website and select a Google Account to login with. Then if your Google account is already associated with an owner entry, then an entry will not be created. Otherwise, a new owner entry will be created with your JWT token's sub value. This sub value will be the owner's uniqueID property.

Get Owners

Allows you to get all existing users

GET /users

Request

Request Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	406 Not Acceptable	If client request the response to be other, non-application/json data.

Response Examples

Success

Status: 200 OK

```
{
  "uniqueID": " 108240556924460704143",
  "id": "5636645067948032"
}
```

Failure

Status: 406 Not Acceptable

```
{
  "Error": "Server only responds with application/json data"
}
```

Comment

- Sending a put, patch, post or delete request will result in a 405 Method Not Allowed response.

Failure

Status: 405 Method Not Allowed

```
{
  "Error": "DELETE/PUT/PATCH/POST request not accepted at current route."
}
```

Create a Boat

Allows you to create a new boat under an owner.

POST /boats

Request

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.
Failure	401 Unauthorized	If client did not provide a valid JWT.
Failure	406 Not Acceptable	If client request the response to be other, non-application/json data.
Failure	415 Unsupported Media Type	If request body format isn't application/json data.

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. This value needs to be sent in the response body as shown in the example.
- 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.
- The `owner` attribute will contain the sub value of a valid JWT token.

Success

Status: 201 Created

```
{
  "id": "abc123",
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "self": "http://<your-app>/boats/abc123"
}
```

Failure

Status: 400 Bad Request

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

Status: 401 Unauthorized

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

Status: 406 Not Acceptable

```
{
  "Error": "Server only responds with application/json data"
}
```

Status: 415 Unsupported Media Type

```
{
  "Error": "Server only accepts application/json data"
}
```

Comment

- Requester needs to submit a Bearer token with request.
- Unspecified attributes will be ignored.
- Sending a put, patch or delete request will result in a 405 Method Not Allowed response.

Failure

Status: 405 Method Not Allowed

```
{
  "Error": "DELETE/PUT/PATCH request not accepted at current route."
}
```

Get a Boat

Allows you to get an existing boat tied to an owner

GET /boats/:boat_id

Request

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	401 Unauthorized	If client did not provide a valid JWT.
Failure	403 Forbidden	If client does not have access to the boat (JWT token's sub value does not match boat's owner).
Failure	404 Not Found	No boat with this boat_id exists
Failure	406 Not Acceptable	If client request the response to be other, non-application/json or text/html data.

Response Examples

Success

Status: 200 OK
<pre>{ "id": "abc123", "name": "Sea Witch", "type": "Catamaran", "length": 28, "owner": "108240556924460704143", "loads": [{ "id": "5629654941564928", "self": "http://<your-app>/loads/5629654941564928" }, { "id": "6205318602162176", "self": "http://<your-app>/loads/6205318602162176" }], "self": "http://<your-app>/boats/abc123" }</pre>

Failure

Status: 401 Unauthorized

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

Status: 403 Forbidden

```
{  
  "Error": "Boat access denied"  
}
```

Status: 404 Not Found

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

Status: 406 Not Acceptable

```
{  
  "Error": "Server only responds with application/json data"  
}
```

Comment

- Requester needs to submit a Bearer token with request.
- Sending a post request will result in a 405 Method Not Allowed response.

Failure

Status: 405 Method Not Allowed

```
{  
  "Error": "POST request not accepted at current route."  
}
```

List all Boats

List all the boats tied to an owner.

GET /boats

Request

Request Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	If client did not provide a valid JWT.
Failure	406 Not Acceptable	If client request the response to be other, non-application/json data.

Response Examples

Success

Status: 200 OK

```
{
  "boats": [
    {
      "length": 99,
      "type": "Yatch",
      "owner": "108240556924460704143",
      "name": "Sea Witch",
      "loads": [],
      "id": "4906279973683200",
      "self": "http://<your-app>/boats/4906279973683200"
    },
    {
      "owner": "108240556924460704143",
      "name": "Odyssey",
      "loads": [],
      "length": 99,
      "type": "Yatch",
      "id": "5755374237908992",
      "self": " http://<your-app>/boats/5755374237908992"
    },
    {
      "owner": "108240556924460704143",
      "name": "Sea Maiden II",
      "loads": [],
```

```

    "length": 1450,
    "type": "Cruise",
    "id": "5935110029312000",
    "self": " http://<your-app>/boats/5935110029312000"
  },
  {
    "name": "Sea Maiden",
    "loads": [],
    "length": 1450,
    "type": "Cruise",
    "owner": "108240556924460704143",
    "id": "5988102107561984",
    "self": " http://<your-app>/boats/5988102107561984"
  },
  {
    "owner": "108240556924460704143",
    "name": "Titanic",
    "loads": [],
    "length": 999,
    "type": "Cruise",
    "id": "6221596528214016",
    "self": http://<your-app>/boats/6221596528214016"
  }
],
"next": " http://<your-app>/boats?cursor=CjASKmoVbX5jcy00OTMtZmluYWwtMjc5MjAychELEgRCb2F0GICAgJil0IYLDgAA%3D",
"total_count": 6
}

```

Failure

Status: 401 Unauthorized

```

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

```

Status: 406 Not Acceptable

```

{
  "Error": "Server only responds with application/json data"
}

```

Comment

- Requester needs to submit a Bearer token with request.
- “boats” is an returned array of existing boat objects that are tied to the owner.
- “next” contains a link to the next five existing boat objects.
- “total_count” contains the total number of boats that are tied to the owner.
- Sending a put, patch or delete request will result in a 405 Method Not Allowed response.

The query will return at most five boats. If there are more than five boats on record, it will display a next link where it will direct you to the next five boats' records. The query with the final boat entry will not display the next link. Query will display load information within the boats object if one exists.

Failure

Status: 405 Method Not Allowed

```
{  
  "Error": "DELETE/PUT/PATCH request not accepted at current route."  
}
```

Edit a Boat (PUT)

Allows you to edit a boat tied to an owner.

PUT /boats/:boat_id

Request

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	303 See Other	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes.
Failure	401 Unauthorized	If client did not provide a valid JWT.
Failure	403 Forbidden	If client does not have access to the boat (JWT token's sub value does not match boat's owner).
Failure	404 Not Found	No boat with this boat_id exists
Failure	406 Not Acceptable	If client request the response to be other, non-application/json data.
Failure	415 Unsupported Media Type	If request body format isn't application/json data.

Response Examples

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

Success

Status: 303 See Other

```
{  
  "url": "http://<your-app>/boats/abc123"  
}
```

Failure

Status: 400 Bad Request

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

Status: 401 Unauthorized

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

Status: 404 Not Found

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

Status: 403 Forbidden

```
{  
  "Error": "Boat access denied"  
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "Server only responds with application/json data"  
}
```

Status: 415 Unsupported Media Type

```
{  
  "Error": "Server only accepts application/json data"  
}
```

Comment

- Any attribute listed in the request body that is not name, type or length will be ignored.
- Request submitter must include a valid JWT token that is associated with the boat's owner to make the update.

Edit a Boat (PATCH)

Allows you to edit a boat tied to an owner.

PATCH /boats/:boat_id

Request

Request Parameters

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

Request Body

Required

Request Body Format

JSON

Request JSON Attributes (Require one or more)

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",
  "length": 99
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If none of the required attribute(s) is in the request body.
Failure	401 Unauthorized	If client did not provide a valid JWT.
Failure	403 Forbidden	If client does not have access to the boat (JWT token's sub value does not match boat's owner).
Failure	404 Not Found	No boat with this boat_id exists
Failure	406 Not Acceptable	If client request the response to be other, non-application/json data.
Failure	415 Unsupported Media Type	If request body format isn't application/json data.

Response Examples

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

Success

Status: 200 OK

```
{
  "id": "abc123",
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 99,
  "owner": "108240556924460704143",
  "loads": [],
  "self": "http://<your-app>/boats/abc123"
}
```

Failure

Status: 400 Bad Request

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

Status: 401 Unauthorized

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

Status: 403 Forbidden

```
{
  "Error": "Boat access denied"
}
```

Status: 404 Not Found

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

Status: 406 Not Acceptable

```
{
  "Error": "Server only responds with application/json data"
}
```

Status: 415 Unsupported Media Type

```
{
  "Error": "Server only accepts application/json data"
}
```

Comment

- Any attribute listed in the request body that is not name, type or length will be ignored.

- If the attribute and property is not included in request, the attribute will be untouched in the datastore.
- If all three attributes of boat are provided, it will be treated as a PUT request and updates all three attribute values.
- Request submitter must include a valid JWT token that is associated with the boat's owner to make the update.

Delete a Boat

Allows you to delete a boat. Note that if the boat is loaded, deleting the boat will “unload” the loads.

```
DELETE /boats/:boat_id
```

Request

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	401 Unauthorized	If client did not provide a valid JWT.
Failure	403 Forbidden	If client does not have access to the boat (JWT token's sub value does not match boat's owner).
Failure	404 Not Found	No boat with this boat_id exists

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 401 Unauthorized
```

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

```
Status: 403 Forbidden
```

```
{
  "Error": "Boat access denied"
}
```

```
Status: 404 Not Found
```

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

Comment

- Request submitter must include a valid JWT token that is associated with the boat's owner to make the deletion.

Create a Load

Allows you to create a new load.

POST /loads

Request

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	Delivery date of the load.	Yes

Request Body Example

```
{
  "weight": 1,
  "content": "Cotton",
  "delivery_date": "1/1/2021"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is missing the required attributes, the load must not be created, and 400 status code must be returned.
Failure	406 Not Acceptable	If client request the response to be other, non-application/json data.
Failure	415 Unsupported Media Type	If request body format isn't application/json data.

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. This value needs to be sent in the response body as shown in the example.
- The value of the attribute `self` is a live link to the REST resource corresponding to this load. The link is not stored in datastore.

Success

Status: 201 Created

```
{
  "id": "5073695114526720",
  "weight": 1,
  "content": "Cotton",
  "delivery_date": "1/1/2021",
  "carrier": [],
  "self": "http://<your-app>/loads/5073695114526720"
}
```

Failure

Status: 400 Bad Request

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

Comment

- Unspecified attributes will be ignored.
- Sending a put, patch or delete request will result in a 405 Method Not Allowed response.
- Requester do not need to provide token in request header as Load entity is not tied to a user.

Failure

Status: 405 Method Not Allowed

```
{
  "Error": "DELETE/PUT/PATCH request not accepted at current route."
}
```

Get a Load

Allows you to get a load

GET /loads/:load_id

Request

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 Not Acceptable	If client request the response to be non-application/json data.

Response Examples

Success

```
Status: 200 OK

{
  "id": 6268758825893888,
  "weight": 1,
  "content": "Cotton",
  "delivery_date": "1/1/2021",
  "carrier": [],
  "self": "http://<your-app>/loads/6268758825893888"
}
OR when it is loaded
{
  "id": 5629654941564928,
  "weight": 1,
  "content": "Silk",
  "delivery_date": "1/1/2021",
  "carrier": [
    {
      "boat_id": "6216585006022656",
      "self": "http://<your-app>/boats/6216585006022656"
    }
  ],
  "self": "http://<your-app>/loads/5629654941564928"
}
```

Failure

Status: 404 Not Found

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

Status: 406 Not Acceptable

```
{  
  "Error": "Server only responds with application/json data"  
}
```

Comment

- Sending a post request will result in a 405 Method Not Allowed response.

Failure

Status: 405 Method Not Allowed

```
{  
  "Error": "POST request not accepted at current route."  
}
```

List all Loads

List all the loads.

GET /loads

Request

Request Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	If client request the response to be non-application/json data.

Response Examples

Success

```
Status: 200 OK

{
  "loads": [
    {
      "carrier": [],
      "content": "Iphone",
      "weight": 1000,
      "delivery_date": "9/10/2020",
      "id": "5081054809423872",
      "self": "http://<your-app>/loads/5081054809423872"
    },
    {
      "delivery_date": "9/10/2020",
      "carrier": [],
      "content": "Iphone X",
      "weight": 1000,
      "id": "5095696621371392",
      "self": "http://<your-app>/loads/5095696621371392"
    },
    {
      "weight": 1,
      "delivery_date": "1/1/2021",
      "carrier": [],
      "content": "Silk",
      "id": "5679095853613056",
      "self": "http://<your-app>/loads/5679095853613056"
    }
  ]
}
```



```

    },
    {
      "weight": 100,
      "delivery_date": "9/10/2020",
      "carrier": [],
      "content": "Serum X",
      "id": "5706627130851328",
      "self": "http://<your-app>/loads/5706627130851328"
    },
    {
      "weight": 10,
      "delivery_date": "1/10/2021",
      "carrier": [],
      "content": "Banana",
      "id": "6205318602162176",
      "self": "http://<your-app>/loads/6205318602162176"
    }
  ],
  "next": "http://<your-app>/loads?cursor=CjASKmoVbX5jcy00OTMtZmluYWwtMjc5MjAychELEgRMb2FkGICAgJjF9oILDkgAA%3D",
  "total_count": 6
}

```

Failure

Status: 406 Not Acceptable

```

{
  "Error": "Server only responds with application/json data"
}

```

Comment

- “loads” is an returned array of existing load objects.
- “next” contains a link to the next five existing load objects.
- “total_count” contains the total number of loads in the database.
- Sending a put, patch or delete request will result in a 405 Method Not Allowed response.

The query will return at most five loads. If there are more than five loads on record, it will display a next link where it will direct you to the next five loads’ records. The query with the final load entry will not display the next link. Query will display boat/carrier information within the load object if one exists.

Failure

Status: 405 Method Not Allowed

```

{
  "Error": "DELETE/PUT/PATCH request not accepted at current route."
}

```

Edit a Load (PUT)

Allows you to edit a load.

PUT /loads/:load_id

Request

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	Integer	The weight of the load.	Yes
content	String	The content of the load.	Yes
delivery_date	String	Delivery date of the load.	Yes

Request Body Example

```
{
  "weight": 2,
  "content": "Cotton",
  "delivery_date": "1/1/2021"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	303 See Other	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes.
Failure	404 Not Found	No load with this load_id exists
Failure	406 Not Acceptable	If client request the response to be non-application/json data.
Failure	415 Unsupported Media Type	If request body format isn't application/json data.

Response Examples

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

Success

```
Status: 303 See Other

{
```

```
"url": "http://<your-app>/loads/5073695114526720"
}
```

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": "Server only responds with application/json data"
}
```

Status: 415 Unsupported Media Type

```
{
  "Error": "Server only accepts application/json data"
}
```

Comment

- Any attribute listed in the request body that is not weight, content or delivery_date will be ignored.

Edit a Load (PATCH)

Allows you to edit a load.

PATCH /loads/:load_id

Request

Request Parameters

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

Request Body

Required

Request Body Format

JSON

Request JSON Attributes (Require one or more)

Name	Type	Description	Required?
weight	Integer	The weight of the load.	Yes
content	String	The content of the load.	Yes
delivery_date	String	Delivery date of the load.	Yes

Request Body Example

```
{
  "weight": 10,
  "content": "Cotton"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If none of the required attribute(s) is in the request body.
Failure	404 Not Found	No load with this load_id exists
Failure	406 Not Acceptable	If client request the response to be non-application/json data.
Failure	415 Unsupported Media Type	If request body format isn't application/json data.

Response Examples

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

Success

```
Status: 200 OK

{
  "id": "abc123",
```

```
{
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 99,
  "owner": "108240556924460704143",
  "loads": [],
  "self": "http://<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 load with this load_id exists"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Server only responds with application/json data"
}
```

Status: 415 Unsupported Media Type

```
{
  "Error": "Server only accepts application/json data"
}
```

Comment

- Any attribute listed in the request body that is not weight, content or delivery_date will be ignored.
- If the attribute and property is not included in request, the attribute will be untouched in the datastore.
- If all three attributes of load are provided, it will be treated as a PUT request and updates all three attribute values.

Delete a Load

Allows you to delete a load. Note that if the load is currently on a boat, it will be removed from the boat. The boat will not be deleted.

DELETE /loads/:load_id

Request

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

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

Load Gets Added to Boat

Load is loaded onto boat.

PUT /boats/boat_id/loads/load_id

Request

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 loaded on a boat and that the requester is associated with the boat's owner.
Failure	401 Unauthorized	If client did not provide a valid JWT.
Failure	403 Forbidden	The load is already on another boat or if client does not have access to the boat (JWT token's sub value does not match boat's owner).
Failure	404 Not Found	No boat with this boat_id exists, and/or no load with this load_id exists.

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Unauthorized <pre>{ "Error": "Missing or invalid JWT" }</pre>
--

Status: 403 Forbidden <pre>{ "Error": " The load is already on another boat" }</pre>

Status: 403 Forbidden

```
{
  "Error": "Boat access denied"
}
```

Status: 404 Not Found

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

Comment

- A load cannot be loaded onto more than one boat.
- Load information is added to boat's (loads) information and the load now has the boat's (carrier) information.
- Request submitter must include a valid JWT token that is associated with the boat's owner to make the update.
- Sending a get, post, patch request will result in a 405 Method Not Allowed response.

Failure

Status: 405 Method Not Allowed

```
{
  "Error": " GET/POST/PATCH request not accepted at current route."
}
```


Load Gets Removed From Boat

Load is removed from boat but load entry is not deleted from Load entity.

DELETE /boats/boat_id/loads/load_id

Request

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 is loaded on the boat, and that the requester is associated with the boat's owner.
Failure	401 Unauthorized	If client did not provide a valid JWT.
Failure	403 Forbidden	The load is not on this boat or if client does not have access to the boat (JWT token's sub value does not match boat's owner).
Failure	404 Not Found	No boat with this boat_id exists, and/or no load with this load_id exists.

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Unauthorized

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

Status: 403 Forbidden

```
{
  "Error": "The load is not on this boat"
}
```

Status: 403 Forbidden

```
{  
  "Error": "Boat access denied"  
}
```

Status: 404 Not Found

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

Comment

- Sending a get, post, patch request will result in a 405 Method Not Allowed response.
- Request submitter must include a valid JWT token that is associated with the boat's owner to make the deletion.

Failure

Status: 405 Method Not Allowed

```
{  
  "Error": " GET/POST/PATCH request not accepted at current route."  
}
```