

Final Project: API Documentation

CS 493: Cloud Application Development

Spring 2021

Oregon State University

Paul Newling

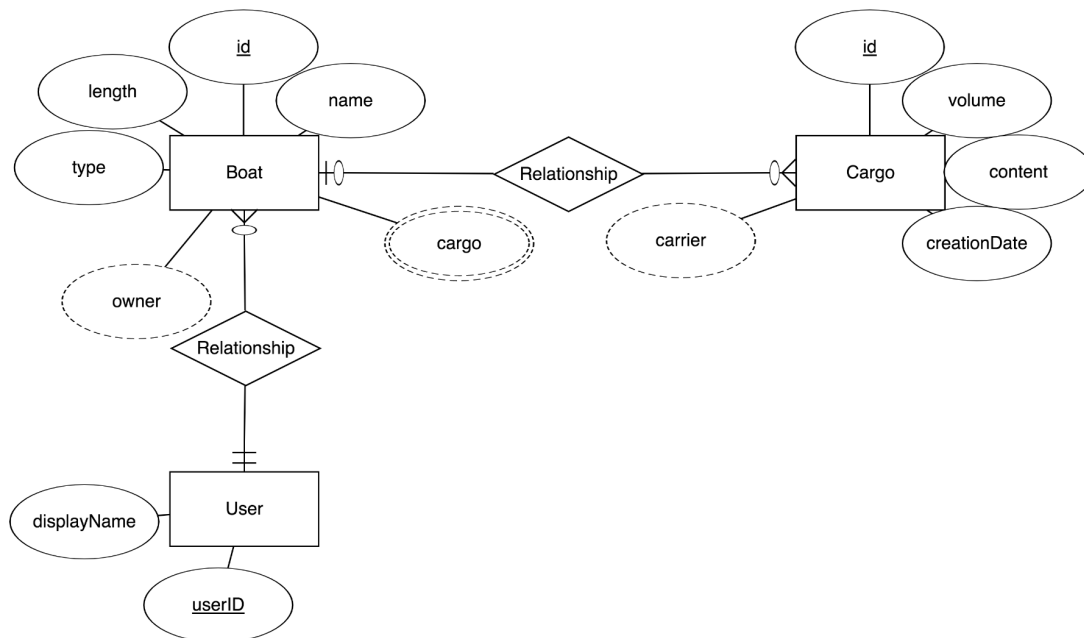
<https://final-newlingp.wl.r.appspot.com/>

Relationships	2
General Notes	3
BOATS	
Create a Boat	6
Delete a Boat	9
Edit a Boat (PATCH)	11
Edit a Boat (PUT)	14
View a Boat	17
View All Boats	19
CARGO	
Create Cargo	22
Delete Cargo	24
Edit Cargo (PATCH)	26
Edit a Cargo (PUT)	28
View a Specific Cargo	31
View All Cargo	33
BOATS & CARGO	
Load Cargo on a Boat	37
Unload Cargo from a Boat	40
USERS	
View All Users	42

Relationships

There are three entities in this project: Users, Boats and Cargo.

- Cargo can optionally be related to one and only one boat (denoted by the carrier attribute).
- A boat can optionally be related to zero or many cargo entities (denoted by the cargo attribute).
- A boat is related to one and only one user (denoted by the owner attribute).
- A user can have zero or many boats



General Notes

JSON Web Tokens (JWT)

Users are required to log in with their Google account in order to generate a valid JWT. On the user's first login their account data and userID will also be saved to the Users datastore. These JWTs are then required for authorization as a bearer token when creating, reading, updating or deleting any of the boat entities. Each boat entity is thus tied to its creator, whose ID it lists under the 'owner' property. Boats created by one user cannot be viewed or modified by a different user. Any attempts to access another user's boat entities is forbidden.

Response Statuses

These are the general responses and status codes given. They can occur when a route doesn't support a certain API method, with certain content-type requests or accepted content-types are not available or accepted.

Status Code	Notes	Example
400 Bad Request	<p>This occurs when a request containing malformed JSON, does not contain all of the required attributes.</p> <p>The response will also return a JSON body:</p> <pre>{ "Error": "Invalid or missing JWT" }</pre>	<p>Using a POST method on the /boats/ route that does not contain the 'length' attribute.</p> <p>Sending malformed JSON. See example below.</p>
401 Unauthorized	<p>This occurs when a request contains a missing or invalid JWT.</p> <p>The response will also return a JSON body:</p> <pre>{ "Error": "The request object is missing at least one of the required attributes" }</pre>	<p>Using a POST method on the /boats/ route that does not contain a valid JWT</p>
403 Forbidden	<p>This occurs when a request is made to modify a boat or boat relationship that does not belong to the user of the JWT</p> <p>The response will also return a JSON body:</p> <pre>{ "Error": "You do not have access to modify or view this boat" }</pre>	<p>Sending a PATCH request to a boat entity that has a different owner than the userID associated with the current JWT</p>

404 Not Found	<p>This occurs when a corresponding entity is not found with the id provided or the JWT is not associated with the entity it is attempting to access.</p> <p>A JSON error message will also be returned.</p>	<p>Using a GET method on the /boats/:boat_id route where the boat_id does not correspond to a valid boat. Or where the boat of boat_id was created by a different user than is encoded in the JWT provided.</p>
405 Method Not Allowed	<p>This occurs when a method is used that is not supported.</p> <p>The 'Accept' header in the response will list the accepted methods available.</p>	<p>Using a DELETE method on the /users/ route.</p> <p>See example below this table.</p>
406 Not Acceptable	<p>This occurs when a request contains an unsupported 'Accept' header</p> <p>The response will return with a header: 'Server Only Responds in JSON'</p>	<p>Using a PATCH method to /boats/:boat_id with a 'Accepts' client header of 'text/html' instead of the accepted 'application/json'.</p> <p>All requests that warrant a response body will be checked to see if they accept application/JSON before the request is processed.</p>
415 Unsupported Media Type	<p>This occurs when a request contains an unsupported media type in the 'content-type' request header.</p> <p>The response will return also return a JSON body:</p> <pre>{ "Error": "Server only accepts application/json data." }</pre>	<p>Using a POST method to /boats/ with a 'content-type' client header of 'text/html' instead of the accepted 'application/json'.</p>

Status 405 Examples

Method and route attempted

DELETE /users

Response

Status 405 Method Not Allowed
Accept: GET

Malformed JSON

Malformed JSON sent, notice extra comma

POST /boats

```
{  
  "name": "Big Bertha",  
  "type": "Frigate",  
  "length": 1100,  
}
```

Response

Status 400 Bad Request

```
{  
  "Error": "Unable to parse JSON"  
}
```

Boats

Boats General Notes

Boat entities are tied to the user that created them. This information is pulled from the JWT that is used in the request. Each boat, in addition to the described attributes below will also contain the property 'cargo'. The cargo property describes which cargo loads are associated with the boat. Upon boat deletion all cargo will be dissociated with it in addition to upon cargo deletion the boat is disassociated with the deleted cargo. In the current iteration there is no limit to the amount of cargo a boat can be associated with.

Create a Boat

Allows you to create a boat.

POST /boats

Request

Path Parameters

None

Request Header

Required - Authorization: Bearer <Valid JWT>

Request Body

Required

Request Body Format

application/JSON

Request JSON Attributes

Name	Type	Description	Required?	Notes
name	String	The name of the boat.	Yes	
type	String	The type of boat.	Yes	
length	Integer	Length of boat in meters	Yes	

Request Body Example

```
{
  "name": "Evergiven",
  "type": "Cargo Freighter",
  "length": 12000
}
```

Response

Response Body Format

application/JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	The response body will contain all of the data associated with the new entry as well as a 'self' attribute that is not stored in the datastore, but links to the REST resource corresponding to this boat.
Failure	400 Bad Request	See General Notes section for more information on this response
Failure	401 Unauthorized	See General Notes section for more information on this response
Failure	406 Not Acceptable	See General Notes section for more information on this response

Examples

Success

```
Status: 201 Created

{
  "id": "5632499082330112",
  "name": "Evergiven",
  "type": "Cargo Freighter",
  "length": 12000,
  "self": "https://<your-app>/boats/5632499082330112"
}
```

Failure

Status: 400 Bad Request

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

Status: 401 Unauthorized

```
{  
  "Error": "Invalid or missing JWT"  
}
```


Delete a Boat

Allows you to delete a boat.

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

Request

Path Parameters

Name	Type	Description
boat_id	String	ID of the boat

Request Header

Required - Authorization: Bearer <Valid JWT>

Request Body

None

Request Body Format

Not applicable

Response

Response Body Format

application/JSON

Success: No body

Failure: application/json

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	See General Notes section for more information on this response
Failure	404 Not Found	See General Notes section for more

		information on this response
Failure	406 Not Acceptable	See General Notes section for more information on this response

Examples

Success

Status: 204 No Content

Failure

Status: 401 Unauthorized

```
{
  "Error": "Invalid or missing JWT"
}
```

Status: 404 Not Found

```
{
  "Error": "No boat with this boat_id exists or you do not have access to it"
}
```

Edit a Boat (PATCH)

Allows you to edit elements of a boat.

PATCH /boats/:boat_id

Request

Path Parameters

Name	Type	Description
boat_id	String	ID of the boat

Request Header

Required - Authorization: Bearer <Valid JWT>

Request Body

Required

Request Body Format

application/JSON

Request JSON Attributes

Name	Type	Description	Required?	Notes
name	String	The name of the boat.	No	
type	String	The type of boat.	No	
length	Integer	Length of boat in meters	No	

Request Body Example

```
{
  "type": "Dingy",
  "length": 4
}
```

Response

Response Body Format

application/JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	The response body will contain all of the data associated with the updated entry as well as a 'self' attribute that is not stored in the datastore, but links to the REST resource corresponding to this boat.
Failure	401 Unauthorized	See General Notes section for more information on this response
Failure	403 Forbidden	See General Notes section for more information on this response
Failure	404 Not Found	See General Notes section for more information on this response
Failure	406 Not Acceptable	See General Notes section for more information on this response

Examples

Success

Status: 200 OK <pre>{ "id": "5632499082330112", "name": "Evergiven", "type": "Dingy", "length": 4, "self": "https://<your-app>/boats/5632499082330112" }</pre>
--

Failure Examples

Status: 401 Unauthorized <pre>{ "Error": "Invalid or missing JWT" }</pre>

```
}
```

Status: 403 Forbidden

```
{
```

```
  "Error": "You do not have access to modify or view this boat"
```

```
}
```

Status: 404 Not Found

```
{
```

```
  "Error": "No boat with this boat_id exists or you do not have access to it"
```

```
}
```

Edit a Boat (PUT)

Allows you to replace boat entry with all new data.

PUT /boats/:boat_id

Request

Path Parameters

Name	Type	Description
boat_id	String	ID of the boat

Request Header

Required - Authorization: Bearer <Valid JWT>

Request Body

Required

Request Body Format

application/JSON

Request JSON Attributes

Name	Type	Description	Required?	Notes
name	String	The name of the boat.	Yes	
type	String	The type of boat.	Yes	
length	Integer	Length of boat in meters	Yes	

Request Body Example

```
{
  "name": "Evergiven",
  "type": "Dingy",
  "length": 4
}
```

Response

Response Body Format

application/JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	The response body will contain all of the data associated with the entry as well as a 'self' attribute that is not stored in the datastore, but links to the REST resource corresponding to this boat.
Failure	400 Bad Request	See General Notes section for more information on this response
Failure	401 Unauthorized	See General Notes section for more information on this response
Failure	403 Forbidden	See General Notes section for more information on this response
Failure	406 Not Acceptable	See General Notes section for more information on this response

Examples

Success

Status: 201 Created

```
{
  "id": "5632499082330112",
  "name": "Evergiven",
  "type": "Cargo Freighter",
  "length": 12000,
  "self": "https://<your-app>/boats/5632499082330112"
}
```

Failure

Status: 400 Bad Request

```
{
```

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

Status: 401 Unauthorized

```
{
  "Error": "Invalid or missing JWT"
}
```

Status: 403 Forbidden

```
{
  "Error": "You do not have access to modify or view this boat"
}
```

Status: 404 Not Found

```
{
  "Error": "No boat with this boat_id exists or you do not have access to it"
}
```


View a Specific Boat

Allows you get information about a boat

GET /boats/:boat_id

Request

Path Parameters

Name	Type	Description
boat_id	String	ID of the boat

Request Header

Required - Authorization: Bearer <Valid JWT>

Request Body

N/A

Response

Response Body Format

application/JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	The response body will contain the specified boat data in the format requested
Failure	401 Unauthorized	See General Notes section for more information on this response
Failure	404 Not Found	See General Notes section for more information on this response
Failure	406 Not Acceptable	See General Notes section for more information on this response

Examples

Success

Status: 200 OK

```
{  
  "id": "5632499082330112",  
  "name": "Evergiven",  
  "type": "Cargo Freighter",  
  "length": 12000,  
  "self": "https://<your-app>/boats/5632499082330112"  
}
```

Failure Examples

Status: 404 Not Found

```
{  
  "Error": "No boat with this boat_id exists or you do not have access to it"  
}
```

View All Boats

Allows you get information about all boats associated with the JWT owner

GET /boats/

Request

Path Parameters

None

Request Header

Required - Authorization: Bearer <Valid JWT>

Request Body

N/A

Response

Response Body Format

application/JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	The boat list uses pagination and if there are more than 5 boats a link will be included to display the remaining boats, with another next link if applicable. There is also a count property at the end of the list that displays the count of boats associated with the JWT owner.
Failure	401 Unauthorized	See General Notes section for more information on this response
Failure	406 Not Acceptable	See General Notes section for more information on this response

Examples

Success

Status: 200 OK

```
{
  "boats": [
    {
      "id": "4810949416452096",
      "name": "Boat1",
      "type": "Skiff",
      "length": 40,
      "owner": "105622306378518676498",
      "cargo": [],
      "self": "https://<your-app>/boats/4810949416452096"
    },
    {
      "id": "5164016397713408",
      "name": "Boat2",
      "type": "Cruiser",
      "length": 400,
      "owner": "105622306378518676498",
      "cargo": [],
      "self": "https://<your-app>/boats/5164016397713408"
    },
    {
      "id": "5726966351134720",
      "name": "Boat3",
      "type": "Tanker",
      "length": 1200,
      "owner": "105622306378518676498",
      "cargo": [],
      "self": "https://<your-app>/boats/5726966351134720"
    },
    {
      "id": "5936849323294720",
      "name": "Boat4",
      "type": "Sailboat",
      "length": 13,
      "owner": "105622306378518676498",
      "cargo": [],
      "self": "https://<your-app>/boats/5936849323294720"
    },
    {
      "id": "6238544469164032",
      "name": "Boat5",
      "type": "Dingy",

```

```
"length": 2,  
"owner": "105622306378518676498",  
"cargo": [],  
"self": "https://<your-app>/boats/6238544469164032"  
}  
],  
"next":  
"https://<your-app>/boats?cursor=CiwSJmoQbX5maW5hbC1uZXdsaW5ncHISCxIFQm9hdHMYglCAmMW9igsMGAAGAA=",  
"count": 6  
}
```

Failure Examples

Status: 401 Unauthorized

```
{  
  "Error": "Invalid or missing JWT"  
}
```

Cargo

General Cargo Notes

Cargo, unlike boats, are not tied to a user and therefore can be created without a JWT. Beside the attributes given below cargo entities also contain the property 'carrier' which denotes which boat is associated with the cargo. Cargo can only be associated with one boat at a time.

Create Cargo

Allows you to create a cargo entity.

POST /cargo

Request

Path Parameters

None

Request Body

Required

Request Body Format

application/JSON

Request JSON Attributes

Name	Type	Description	Required?	Notes
volume	Integer	The volume of cargo	Yes	
content	String	The content of cargo.	Yes	
creationDate	String	Creation date of cargo	Yes	

Request Body Example

```
{
  "volume": 14,
  "content": "Legos",
  "creationDate": "2021-04-21"
}
```

Response

Response Body Format

application/JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	The response body will contain all of the data associated with the new entry as well as a 'self' attribute that is not stored in the datastore, but links to the REST resource corresponding to this cargo.
Failure	400 Bad Request	See General Notes section for more information on this response
Failure	406 Not Acceptable	See General Notes section for more information on this response

Examples

Success

Status: 201 Created

```
{
  "id": "6468260526555136",
  "volume": 14,
  "carrier": null,
  "content": "Legos",
  "creationDate": "2021-04-21",
  "self": "https://<your-app>/cargo/5632499082330112"
}
```

Failure

Status: 400 Bad Request

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

Delete Cargo

Allows you to delete a cargo entity.

```
DELETE /cargo/:cargo_id
```

Request

Path Parameters

Name	Type	Description
cargo_id	String	ID of the cargo

Request Body

None

Request Body Format

Not applicable

Response

Response Body Format

application/JSON

Success: No body

Failure: application/json

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	404 Not Found	See General Notes section for more information on this response
Failure	406 Not Acceptable	See General Notes section for more information on this response

Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found

```
{  
  "Error": "No cargo with this cargo_id exists"  
}
```

Edit Cargo (PATCH)

Allows you to edit elements of a cargo entity.

PATCH /cargo/:cargo_id

Request

Path Parameters

Name	Type	Description
cargo_id	String	ID of the cargo

Request Body

Required

Request Body Format

application/JSON

Request JSON Attributes

Name	Type	Description	Required?	Notes
volume	Integer	The volume of cargo	No	
content	String	The content of cargo.	No	
creationDate	String	Creation date of cargo	No	

Request Body Example

```
{
  "volume": 12,
  "content": "Rocks"
}
```

Response

Response Body Format

application/JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	The response body will contain all of the data associated with the updated entry as well as a 'self' attribute that is not stored in the datastore, but links to the REST resource corresponding to this cargo.
Failure	404 Not Found	See General Notes section for more information on this response
Failure	406 Not Acceptable	See General Notes section for more information on this response

Examples

Success

Status: 201 Created

```
{
  "id": "6468260526555136",
  "volume": 12,
  "carrier": null,
  "content": "Rocks",
  "creationDate": "2021-04-21",
  "self": "https://<your-app>/cargo/5632499082330112"
}
```

Failure Examples

Status: 404 Not Found

```
{
  "Error": "No cargo with this cargo_id exists"
}
```

Edit a Cargo (PUT)

Allows you to replace cargo entry with all new data.

PUT /cargo/:cargo_id

Request

Path Parameters

Name	Type	Description
cargo_id	String	ID of the cargo

Request Body

Required

Request Body Format

application/JSON

Request JSON Attributes

Name	Type	Description	Required?	Notes
volume	Integer	The volume of cargo	Yes	
content	String	The content of cargo.	Yes	
creation Date	String	Creation date of cargo	Yes	

Request Body Example

```
{
  "volume": 14,
  "content": "Legos",
  "creationDate": "2021-04-21"
}
```

Response

Response Body Format

application/JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	The response body will contain all of the data associated with the entry as well as a 'self' attribute that is not stored in the datastore, but links to the REST resource corresponding to this cargo.
Failure	400 Bad Request	See General Notes section for more information on this response
Failure	406 Not Acceptable	See General Notes section for more information on this response

Examples

Success

Status: 201 Created

```
{
  "id": "6468260526555136",
  "volume": 14,
  "carrier": null,
  "content": "Legos",
  "creationDate": "2021-04-21",
  "self": "https://<your-app>/cargo/5632499082330112"
}
```

Failure

Status: 400 Bad Request

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

Status: 404 Not Found

```
{  
  "Error": "No cargo with this cargo_id exists"  
}
```

View a Specific Cargo

Allows you get information about a specific cargo entity.

```
GET /cargo/:cargo_id
```

Request

Path Parameters

Name	Type	Description
cargo_id	String	ID of the cargo

Request Body

N/A

Response

Response Body Format

application/JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	The response body will contain the specified cargo data in the format requested
Failure	404 Not Found	See General Notes section for more information on this response
Failure	406 Not Acceptable	See General Notes section for more information on this response

Examples

Success

Status: 201 Created

```
{
```

```
{
  "id": "6468260526555136",
  "volume": 14,
  "carrier": null,
  "content": "Legos",
  "creationDate": "2021-04-21",
  "self": "https://<your-app>/cargo/5632499082330112"
}
```

Failure

Status: 400 Bad Request

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

Status: 404 Not Found

```
{
  "Error": "No cargo with this cargo_id exists"
}
```


View All Cargo

Allows you to view all cargo

GET /cargo

Request

Path Parameters

None

Request Body

N/A

Response

Response Body Format

application/JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	The cargo list uses pagination and if there are more than 5 cargo entities a link will be included to display the remaining cargo, with another next link if applicable. The count property will also be attached to the response that displays the total count of cargo entities.
Failure	406 Not Acceptable	See General Notes section for more information on this response

Examples

Success

Status: 201 Created

```
{
  "cargo": [
    {
      "id": "4511129829113856",
```

```
"volume": 12,
"carrier": null,
"content": "Rocks",
"creationDate": "2021-04-21",
"self": "https://final-newlingp.wl.r.appspot.com/cargo/4511129829113856"
},
{
  "id": "5475870214979584",
  "volume": 10,
  "carrier": null,
  "content": "Legos",
  "creationDate": "2021-05-20",
  "self": "https://final-newlingp.wl.r.appspot.com/cargo/5475870214979584"
},
{
  "id": "5542103341858816",
  "volume": 102,
  "carrier": null,
  "content": "Cars",
  "creationDate": "2021-01-05",
  "self": "https://final-newlingp.wl.r.appspot.com/cargo/5542103341858816"
},
{
  "id": "6038820168400896",
  "volume": 17,
  "carrier": null,
  "content": "Rice",
  "creationDate": "2020-12-21",
  "self": "https://final-newlingp.wl.r.appspot.com/cargo/6038820168400896"
},
{
  "id": "6668003248701440",
  "volume": 42,
  "carrier": null,
  "content": "Vaccines",
  "creationDate": "2021-02-15",
  "self": "https://final-newlingp.wl.r.appspot.com/cargo/6668003248701440"
}
],
"next":
"https://final-newlingp.wl.r.appspot.com/cargo?cursor=CiwSJmoQbX5maW5hbC1uZXdsaW5ncHISCxIFQ2FyZ28YglCA6LeQ7AsMGAAGAA=",
"count": 6
}
```

Failure

Status: 400 Bad Request

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

Boats and Cargo

Boats and Cargo General Notes

Boat entities are tied to the user that created them. They can only be modified by the userID associated with their creation (which is pulled from the corresponding JWT). This also applies to loading and unloading cargo onto boats. Boats can be associated with zero or many cargo entities. Cargo can be associated with zero or one boat. These rules hold when making or dissolving associations between the two, hereto referred to as 'loading' and 'unloading' cargo onto a boat.

Load Cargo on a Boat

Allows you to associate a cargo entity with a boat.

PATCH /boats/:boat_id/cargo:cargo_id

Request

Path Parameters

Name	Type	Description
boat_id	String	ID of the boat
cargo_id	String	ID of cargo entity

Request Header

Required - Authorization: Bearer <Valid JWT>

Request Body

None

Request Body Format

application/JSON

Request JSON Attributes

N/A

Response

Response Body Format

application/JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	The response body will contain all of the data associated with the boat entry with the new cargo association as well as a 'self' attribute that is not stored in the datastore, but links to the REST resource corresponding to this boat.
Failure	401 Unauthorized	See General Notes section for more information on this response
Failure	403 Forbidden	<p>This response can occur if the cargo is already associated with another boat or if the current JWT userID is not associated with the boat of the boat_id.</p> <p>See General Notes section for more information on this response.</p>
Failure	404 Not Found	See General Notes section for more information on this response
Failure	406 Not Acceptable	See General Notes section for more information on this response

Examples

Success

Status: 201 Created

```
{
  "id": "4810949416452096",
  "name": "Evergiven",
  "type": "Cargo Freighter",
  "length": 12000,
  "owner": "105622306378518676498",
  "cargo": [
    {
```

```
"id": "4511129829113856",  
  "self": "https://<your-app>/cargo/4511129829113856"  
},  
  "self": "https://<your-app>/boats4810949416452096"  
}
```

Failure

Status: 401 Unauthorized

```
{  
  "Error": "Invalid or missing JWT"  
}
```

Status: 403 Forbidden - When cargo entity is already assigned to a boat

```
{  
  "Error": "The specified cargo is already assigned a boat"  
}
```

Status: 403 Forbidden - current JWT userID is not associated with the boat

```
{  
  "Error": "You do not have access to modify or view this boat"  
}
```

Status: 404 Not Found

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

Unload Cargo from a Boat

Allows you to disassociate a cargo entity from a boat.

```
DELETE /boats/:boat_id/cargo:cargo_id
```

Request

Path Parameters

Name	Type	Description
boat_id	String	ID of the boat
cargo_id	String	ID of cargo entity

Request Header

Required - Authorization: Bearer <Valid JWT>

Request Body

None

Request Body Format

application/JSON

Request JSON Attributes

N/A

Response

Response Body Format

application/JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	The response body will contain all of the data associated with the boat entry with the cargo association removed as well as a 'self' attribute that is not stored in the datastore,

		but links to the REST resource corresponding to this boat.
Failure	401 Unauthorized	See General Notes section for more information on this response
Failure	403 Forbidden	This response can occur if the cargo is already associated with another boat or if the current JWT userID is not associated with the boat of the boat_id. See General Notes section for more information on this response.
Failure	404 Not Found	See General Notes section for more information on this response
Failure	406 Not Acceptable	See General Notes section for more information on this response

Examples

Success

```
Status: 201 Created
{
  "id": "4810949416452096",
  "name": "Evergiven",
  "type": "Cargo Freighter",
  "length": 12000,
  "owner": "105622306378518676498",
  "cargo": [ ],
  "self": "https://<your-app>/boats4810949416452096"
}
```

Failure

```
Status: 401 Unauthorized

{
  "Error": "Invalid or missing JWT"
}

Status: 403 Forbidden - When cargo entity is not assigned to the boat
{
  "Error": "The specified boat is not assigned to this cargo"
}
```


Status: 403 Forbidden - current JWT userID is not associated with the boat

```
{  
  "Error": "You do not have access to modify or view this boat"  
}
```

Status: 404 Not Found

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

Users

General Users Notes

Users are added to the datastore after first logging into the main page. After a user has been authenticated they will use the JWT produced to be able to create, read, update and delete boat entities. Each user will be tied to the boat entity by the property 'owner' of each boat. The owner property relates to the userID. User entities cannot be modified or deleted.

View All Users

Allows you view all users

GET /users/

Request

Path Parameters

None

Request Header

None

Request Body

N/A

Response

Response Body Format

application/JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	The boat list uses pagination and if there are more than 5 boats a link will be included to display the remaining boats, with another next link if applicable. A count

		property will also be attached to the response that displays the total number of users in the datastore.
Failure	405 Method Not Allowed	See General Notes section for more information on this response
Failure	406 Not Acceptable	See General Notes section for more information on this response

Examples

Success

Status: 200 OK

```
{
  "users": [
    {
      "userID": "105622306378518676498",
      "displayName": "Paul Newling"
    },
    {
      "userID": "651357564651615347321",
      "displayName": "Joe McJoeJoe"
    },
    {
      "userID": "6548573574651654651361",
      "displayName": "Jane Cloud"
    },
    {
      "userID": "1654357651654324651654",
      "displayName": "Debbie Science"
    },
    {
      "userID": "1654354651654316516546",
      "displayName": "Kyle Computer"
    }
  ],
  "next":
  "https://<your-app>/users?cursor=CiwSJmoQbX5maW5hbC1uZXdsaW5ncHISCxIFQm9hdHM
  YglCAmMW9igsMGAAGAA=",
  "count": 6
}
```

Failure Examples

Status: 405 Method Not Allowed Accept: GET
--