Megan McCarron-Haber
CS493 Final Project
December 7, 2020

This application can be found at: mccarrom-final.ue.r.appspot.com

The user can create an account or login at this same URL

Data Model

Non-User Entities

Boats

Property	Type	Description	Valid values	Required?
name	String	The name of the boat.	“Mary” “Mary Queen of Scotts” “Bob”	Yes
type	String	The type of the boat. E.g., Sailboat, Catamaran, etc.	“Canoe” “Schooner” “Sailboat” “Boat”	Yes
length	Integer	Length of the boat in feet.	1, 2, 99, 87 etc	Yes

Loads

Property	Type	Description	Valid values	Required?
Content	String	What the load is carrying	“Peanut Butter” “Coal” “presents”	Yes
Delivery_date	String	When the load will be delivered	“12/24/2020” “12242020” “24/12/2020” “12/14/20”	Yes
Weight	Integer	How heavy is the load	100, 3, 5,79	Yes

Non-user entity relationship: Each boat can have zero to many loads, each load can belong to zero or one boats.

User entity

Property	Type	Description	Valid values	Required?
UserID	String	Unique Identifier derived from the sub property of the generated JWT for the user	“102285637417655049660” “113297724479619112051”	Yes

Each user owns zero to many boats, which can be carrying zero to many loads. Loads do not directly belong to the user.

The unique identifier for each user is the UserID, which is the sub property of the generated JWT. When the user goes to the index page, they are redirected to a Google Authentication page. The user chooses the appropriate Google account and is redirected back to a page with their User Information. This page displays their UserID and their JWT. When this page is returned the user's UserID is compared with the existing UserIDs in the Datastore. If the user has never authenticated through Google for this application before, their UserID is added to the Datastore. For each request that requires user authentication, the user must supply their JWT to the application.

You can add any number of loads to a boat, but a load can only be on one boat at any given time. If you attempt to add a load to a boat which is already on a different boat, an error will return. You can only add loads to boats you own, and you can only remove loads from boats you own.

If a boat is deleted, any loads that were on it are free to be added to a new boat. If a load is deleted, it is no longer represented in a boat's loads attribute.

API Spec

CS 493: Cloud Application Development

Fall 2020

Oregon State University

Change log

Page	Change	Date
	Initial version.	December 7, 2020

Create a Boat

Allows you to create a new boat.

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": "Moose",
  "type": "Canoe",
  "length": 9
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, the boat must not be created, and 400 status code must

		be returned.
Failure	401 Unauthorized	If the user making the request is missing a valid JWT with their request, the boat must not be created, and a 401 status code must be returned
Failure	403 Forbidden	If the request provides an invalid JWT, the boat must not be created, and a 403 status code must be returned.

Success

```
Status: 201 Created

{
  "id": "abc123",
  "name": "Moose",
  "type": "Canoe",
  "length": 9,
  "self": "https://<your-app>/boats/abc123"
}
```

Failure

```
Status: 400 Bad Request

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

Failure

```
Status: 401 Unauthorized

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

Failure

```
Status: 403 Forbidden

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

Get a Boat

Allows you to get an existing boat

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	404 Not Found	No boat with this boat_id exists
Failure	401 Unauthorized	The requesting user has not proven they are authorized to view this boat
Failure	403 Forbidden	The requesting user is not authorized to view this boat

Response Examples

Success

```
Status: 200 OK

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

Failure

```
Status: 404 Not Found

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

Failure

```
Status: 401 Unauthorized

{
"Error":  "You must be logged in to view this boat (Missing JWT)”
}
```

Failure

```
Status: 403 Forbidden

{
"Error":  "You do not own this boat. You do not have permission to view this boat”
}
```

List all Boats

List all the boats.

```
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	The requesting user has not proven they are authorized to view boats
Failure	403 Forbidden	The requesting user is not authorized to view boats

Response Examples

Success

```
Status: 200 OK

[
{
  "id": "abc123",
  "name": "Sea Witch",
   "type": "Catamaran",
  "length": 28,
  "self": "https://<your-app>/boats/abc123"
},
{
  "id": "def456",
  "name": "Adventure",
   "type": "Sailboat",
  "length": 50,
  "self": "https://<your-app>/boats/def456"
},
{
  "id": "xyz123",
  "name": "Hocus Pocus",
   "type": "Sailboat",
  "length": 100,
  "self": "https://<your-app>/boats/xyz123"
}
]
```

Failure

Status: 401 Unauthorized

```
{
"Error":  "You must be logged in to view boats (Missing JWT)”
}
```

Failure

Status: 403 Forbidden

```
{
"Error":  "You do not have permission to view these boats (Invalid JWT)”
}
```

Edit a Boat

Allows you to edit a boat.

```
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	200 OK	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, the boat must not be updated, and 400 status code must be returned.
Failure	404 Not Found	No boat with this boat_id exists
Failure	401 Unauthorized	The requesting user has not proven they are authorized to edit boats
Failure	403 Forbidden	The requesting user is not authorized to edit these boats

Response Examples

Success

```
Status: 200 OK

{
  "id": "abc123",
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 99,
  "self": "https://<your-app>/boats/abc123"
}
```

Failure

```
Status: 400 Bad Request

{
"Error":  "The request object is missing a required field."
}
```

```
Status: 404 Not Found

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

Failure

```
Status: 401 Unauthorized

{
"Error":  "You must be logged in to edit boats”
}
```

Failure

```
Status: 403 Forbidden

{
"Error":  "You do not have permission to edit this boat”
}
```

Allows you to edit a boat.

```
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

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 Warlock",
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If the fields indicated in the request are not attributes of the boat, the boat must not be updated, and 400 status code must be returned.
Failure	404 Not Found	No boat with this boat_id exists
Failure	401 Unauthorized	The requesting user has not proven they are authorized to edit boats
Failure	403 Forbidden	The requesting user is not authorized to edit these boats

Response Examples

Success

```
Status: 200 OK

{
  "id": "abc123",
  "name": "Sea Warlock",
  "type": "Catamaran",
  "length": 99,
  "self": "https://<your-app>/boats/abc123"
}
```

Failure

```
Status: 400 Bad Request

{
"Error":  "There is something wrong with the request object."
}
```

```
Status: 404 Not Found

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

Failure

```
Status: 401 Unauthorized

{
"Error":  "You must be logged in to edit boats (Missing JWT)”
}
```

Failure

```
Status: 403 Forbidden

{
"Error":  "You do not have permission to edit these boats (Invalid JWT)”
}
```

Delete a Boat

Allows you to delete a boat. Note that if the boat is currently in a load, deleting the boat makes the load empty.

```
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	404 Not Found	No boat with this boat_id exists
Failure	401 Unauthorized	The requesting user has not proven they are authorized to delete boats
Failure	403 Forbidden	The requesting user is not authorized to delete these boats

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 404 Not Found

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

Failure

```
Status: 401 Unauthorized

{
"Error":  "You must be logged in to delete boats (Missing JWT)”
}
```

Failure

```
Status: 403 Forbidden

{
"Error":  "You do not have permission to delete these boats (Invalid JWT)”
}
```

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?
Content	String	What the load is carrying	Yes
Delivery_date	String	When the load will be delivered	Yes
Weight	Integer	How heavy is the load	Yes

Request Body Example

```
{
  "content” : “coal”,
  “delivery_date” : “12/24/2020”,
  “weight” : 99
}
```

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 required attribute, the load must not be created and 400 status must be returned.

Success

```
Status: 201 Created

{
  "id": "123abc",
  "content": coal,
   "weight": 99,
  “delivery_date” : “12/24/2020”,
  "self": "https://<your-app>/loads/123abc"
}
```

Failure

```
Status: 400 Bad Request

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

Get a Load

Allows you to get an existing 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

Response Examples

Success

```
Status: 200 OK

{
  "id": "123abc",
  "content": coal,
   "weight": 99,
  “delivery_date” : “12/24/2020”,
  "self": "https://<your-app>/loads/123abc"
}
```

Failure

```
Status: 404 Not Found

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

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	

Response Examples

Success

```
Status: 200 OK

[
{
  "id": "123abc",
  "content": coal,
   "weight": 99,
  “delivery_date” : “12/24/2020”,
  "self": "https://<your-app>/loads/123abc"
},
{
  "id": "456abc",
  "content": presents,
   "weight": 999,
  “delivery_date” : “12/24/2020”,
  "self": "https://<your-app>/loads/456abc"
}
]
```

Delete a Load

Allows you to delete a load. If the load being deleted has a boat, the boat is now considered “at sea.”

```
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"
}
```

Edit a Load

Allows you to edit a boat.

PUT /loads/:load_id

Request

Request Parameters

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

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
Content	String	What the load is carrying	Yes
Delivery_date	String	When the load will be delivered	Yes
Weight	Integer	How heavy is the load	Yes

Request Body Example

```
{
  "content": "presents",
  "delivery_date": "12/10/2020",
  "weight": 98
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, the load must not be updated, and 400 status code must be returned.
Failure	404 Not Found	No load with this load_id exists

Response Examples

Success

```
Status: 200 OK

{
  "id": "abc123",
  "content": "presents",
  "delivery_date": "12/10/2020",
  "weight": 98
  "self": "https://<your-app>/boats/abc123"
}
```

Failure

```
Status: 400 Bad Request

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

```
Status: 404 Not Found

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

Allows you to edit a boat.

```
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

Name	Type	Description	Required?
Content	String	What the load is carrying	No
Delivery_date	String	When the load will be delivered	No
Weight	Integer	How heavy is the load	No

Request Body Example

```
{
  "delivery_date": "12/10/2020",
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If the fields indicated in the request are not attributes of the load, the load must not be updated, and 400 status code must be returned.
Failure	404 Not Found	No boat with this boat_id exists

Response Examples

Success

```
Status: 200 OK

{
  "id": "123abc",
  "content": coal,
  "weight": 99,
  “delivery_date” : “12/1/2020”,
  "self": "https://<your-app>/loads/123abc"
},
```

Failure

```
Status: 400 Bad Request

{
"Error":  "The attributes in the request are invalid”
}
```

```
Status: 404 Not Found

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

Put a Load on a Boat

Boat has arrived at a load.

```
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 already on a boat
Failure	403 Forbidden	This load is not on this boat
Failure	403 Forbidden	You do not own this boat and therefore cannot add loads to this boat
Failure	401 Unauthorized	Missing JWT or Invalid JWT
Failure	404 Not Found	No boat with this boat_id exists, and/or no load with this load_id exits.

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 403 Forbidden

{
"Error":  "This load is already on another boat"
}
```

Failure

```
Status: 404 Not Found

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

Failure

```
Status: 403 Forbidden

{
"Error":  "You do not own this boat, therefore you cannot add loads to this boat"
}
```

Failure

```
Status: 401 Unauthorized

{
"Error":  "Missing JWT" or “Invalid JWT”
}
```

Remove a load from a boat

Boat has arrived at a load.

```
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 the load is not already on a boat
Failure	403 Forbidden	This load is not on this boat
Failure	403 Forbidden	You do not own this boat and therefore cannot remove loads from this boat
Failure	401 Unauthorized	Missing JWT or Invalid JWT
Failure	404 Not Found	No boat with this boat_id exists, and/or no load with this load_id exits.

Response Examples

Success

Status: 204 No Content

Failure

Status: 403 Forbidden

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

Failure

Status: 404 Not Found

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

Failure

Status: 403 Forbidden

```
{
"Error":  "You do not own this boat, therefore you cannot remove loads from this
boat"
}
```

Failure

Status: 401 Unauthorized

```
{
"Error":  "Missing JWT" or “Invalid JWT”
}
```