

# Kleditzl CS493 Final Project

CS 493: Cloud Application Development

Fall 2022

Oregon State University

<https://final-kleditzl.uw.r.appspot.com/>

## Contents

Data Model .....	1
List all Users (Unprotected) .....	2
Create a Boat (Protected) .....	3
Get a Boat (Protected) .....	6
List all Boats (Protected) .....	8
Patch a Boat (Protected) .....	11
Put a Boat (Protected) .....	13
Delete a Boat (Protected) .....	17
Create a Load (Unprotected) .....	19
Get a Load (Unprotected) .....	21
List all Loads (Unprotected) .....	22
Delete a Load (Unprotected) .....	23
Boat receives a load (Protected) .....	24
Boat removes a Load (Protected) .....	26

## Data Model

The app stores three kinds of entities in Datastore, Boat, Loads, and Users. Anytime you are making a change to the core attributes of a boat, it will require the user to supply a JWT token that matches the owner attribute of the boat.

Boats (name, type, length are all protected in the 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. This must be unique across the datastore.
type	String	Type of the boat. E.g., Sailboat, Catamaran, etc.
length	Integer	The length of the boat in feet.
Loads	List/Array	The list of the loads assigned to the boat. This does not need to be passed in and must be updated with a put function. It

		includes the loads' id and the self-link.
Owner	String	This is the sub portion of the JWT token provided to access the boat content.
Self	String	A string that has the link to an https get request of itself.

### Loads (This data is unprotected)

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 in integers.
item	string	The item that is in the load. E.g., Lego Blocks
Creation_Date	String	The string of the date the load was created. E.g., 10/18/2022
Carrier	List/Array	This is the carrier of the load. This list can only have 1 carrier/boat and will include the boats id and a self-link.
Self	String	A string that has the link to an https get request of itself.

### Users (This data is unprotected)

This cannot be accessed except for a single get request. These are added when users generate their JWT token.

Property	Data Type	Notes
id	Integer	The id of the user. Datastore automatically generates it. Don't add it yourself as a property of the entity.
First_name	String	First name of the user provided by Google.
Last_name	String	Last name of the user provided by Google.
Sub	String	The sub attribute of the JWT token provided by Google's OAuth.

### List all Users (Unprotected)

List all the boats.

GET /users
------------

Request

Path Parameters

None

Authorization Header

None

Request Body

None

## Response

### Response Body Format

JSON

### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

### Response Examples

#### Success

Status: 200 OK
<pre>[   {     "id": "5709113312935936",     "first_name": "Logan Makenzie",     "last_name": "Kleditz",     "sub": "101644885892357229942",   } ]</pre>

## Create a Boat (Protected)

Allows you to create a new boat.

POST /boats
-------------

### Request

#### Path Parameters

None

#### Authorization Header

Bearer JWT

## Response

### Response Body

Required

### Response 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	<p>If the request is missing any of the 3 required attributes, the boat must not be created, and 400 status code must be returned.</p> <p>You don't need to validate the values of the attributes and can assume that if the request contains any of the listed attributes, then that attribute's value is valid.</p> <p>This API does not check for extraneous attributes. They will simply be ignored.</p>
Failure	401 Unauthorized	The header is missing the required JWT token to create a new boat.
Failure	403 Forbidden	The name must be unique.
Failure	406 Not Acceptable	The request had the wrong request type.
Failure	415 Unsupported Media Type	The request has the wrong content-type.

### Response Examples

- Datastore will automatically generate an ID and sub if the JWT token is provided. Then it will be stored in the Google Cloud Data Store.

#### Success

Status: 201 Created

```
{
  "id": 4623203678814208,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "loads": [],
  "self": "https://final-kleditzl.uw.r.appspot.com/boats/4623203678814208"
}
```

#### Failure

Status: 400 Bad Request

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

*Failure*

Status: 401 Unauthorized

```
<!DOCTYPE html>  
<html lang="en">  
  
<head>  
  <meta charset="utf-8">  
  <title>Error</title>  
</head>  
  
<body>  
  <pre>Unauthorized</pre>  
</body>
```

*Failure*

Status: 403 Forbidden

```
{  
  "Error": "The name of the boat must be unique."  
}
```

*Failure*

Status: 406 Not Acceptable

```
{  
  "Error": "Not an acceptable request type."  
}
```

*Failure*

Status: 415 Unsupported Media Type

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

## Get a Boat (Protected)

Allows you to get an existing boat

GET /boats/:boat\_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request

Request Body

None

Authorization Header

Bearer JWT

Response

Response Body

Required

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	The sent JWT token does not match the owner of the boat.
Failure	404 Not Found	No boat with this boat_id exists
Failure	406 Not Acceptable	The request type does not match Accept application/json
Failure	415 Unsupported Media Type	The request has the wrong content-type.

Response Examples

*Success*

Status: 200 OK

```
{
  "id": 123,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28
}
```

*Failure*

Status: 401 Unauthorized

No token provided:

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
  <meta charset="utf-8">
  <title>Error</title>
</head>
```

```
<body>
  <pre>Unauthorized</pre>
</body>
```

Incorrect Token Provided:

```
{"Error": "The token you provided is not authorized to view this boat."}
```

*Failure*

Status: 404 Not Found

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

*Failure*

Status: 406 Not Acceptable

```
{
  "Error": " Not an acceptable request type."
}
```

*Failure*

Status: 415 Unsupported Media Type

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

## List all Boats (Protected)

List all the boats.

GET /boats

Request

Request Body

None

Authorization Header

Bearer JWT

Response

Response Body

Required

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	If there isn't a token provided, then you cannot get even an empty json as a response. A JWT token must be given and verified.
Failure	406 Not Acceptable	The request type does not match Accept application/json
Failure	415 Unsupported Media Type	The request has the wrong content-type.

Response Examples

- This request will only return up to 5 boats for the owner of the boats. If the owner has more than 5 boats in the datastore, it will return a cursor to the next list of up to 5 boats. This response will not return a boat if the JWT token does not match the owner variable of that boat.

*Success*

Status: 200 OK

```
{
  "items": [
    {
      "name": "Sea Witch 3",
      "type": "Catamaran",
      "loads": [],
      "owner": "101644885892357229942",
      "length": 28,
      "id": "4710730112696320"
    },
    {
      "type": "Catamaran",
```



```
    "length": 28,
    "loads": [],
    "name": "Sea Witch 5",
    "owner": "101644885892357229942",
    "id": "5206551136567296"
  },
  {
    "name": "Sea Witch",
    "type": "Catamaran",
    "owner": "101644885892357229942",
    "length": 28,
    "loads": [],
    "id": "5474530185183232"
  },
  {
    "name": "Sea Witch 2",
    "owner": "101644885892357229942",
    "type": "Catamaran",
    "loads": [],
    "length": 28,
    "id": "5699409840963584"
  },
  {
    "length": 28,
    "loads": [],
    "owner": "101644885892357229942",
    "type": "Catamaran",
    "name": "Sea Witch 4",
    "id": "5737525125578752"
  }
],
"next": "https://final-
kleditzl.uw.r.appspot.com/boats?cursor=Ci0SJ2oSenV3fmZpbmFsLWtsZWRpdHpschELEgRCb2F0GICAgNj
6x5gKDBgAIAA="
}
```

#### *Failure*

Status: 401 Unauthorized

```
{"Error": "No token found, or the token provided did not give you clearance to perform this action."}
```

#### *Failure*

Status: 404 Not Found

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

*Failure*

Status: 406 Not Acceptable

```
{  
  "Error": " Not an acceptable request type."  
}
```

*Failure*

Status: 415 Unsupported Media Type

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

## Patch a Boat (Protected)

Allows you to edit a subset of a boats attributes.

PATCH /boats/:boat\_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request

Request Body

None

Authorization Header

Bearer JWT

Response

Response Body

Required

Response 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

<pre>{   "name": "Sea Witch",   "type": "Catamaran" }</pre>
Or
<pre>{   "length": 99 }</pre>

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 will not be updated, and 400 status code is

		<p>returned.</p> <p>The attribute types are not validated, if there are superfluous or incorrect attributes they will be ignored.</p> <p>The name must be unique.</p>
Failure	401 Unauthorized	If there isn't a token provided, then you cannot get even an empty json as a response. A JWT token must be given and verified.
Failure	403 Forbidden	The name must be unique.
Failure	404 Not Found	No boat with this boat_id exists.
Failure	406 Not Acceptable	The request type does not match Accept application/json
Failure	415 Unsupported Media Type	The request has the wrong content-type.

## Response Examples

### Success

Status: 200 OK

```
{
  "id": 123,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 99
}
```

### Failure

Status: 400 Bad Request

```
{"Error": "One of the attributes are in the incorrect type"}
```

### Failure

Status: 401 Unauthorized

No JWT sent:

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
  <meta charset="utf-8">
  <title>Error</title>
</head>
```

```
<body>
  <pre>Unauthorized</pre>
</body>
```

JWT does not match the owner:

```
{"Error": "No token found, or the token provided did not give you clearance to perform this action."}
```

#### Failure

Status: 403 Forbidden

```
{  
  "Error": "The name of the boat must be unique."  
}
```

#### Failure

Status: 404 Not Found

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

#### Failure

Status: 406 Not Acceptable

```
{  
  "Error": "Not an acceptable request type."  
}
```

#### Failure

Status: 415 Unsupported Media Type

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

## Put a Boat (Protected)

Allows you to edit a boat.

PUT /boats/:boat\_id

### Request

#### Path Parameters

Name	Description
boat_id	ID of the boat

### Request

#### Request Body

None

#### Authorization Header

Bearer JWT

### Response

#### Response Body

Required

## Response 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": 99
}
```

## Response

### Response Body Format

#### JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	303 See Other	The response will have a header to the location of the boat with its updated attributes
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, the boat will not be updated, and 400 status code is returned.  The attribute types are not validated, if there are superfluous or incorrect attributes they will be ignored.  The name must be unique.
Failure	401 Unauthorized	If there isn't a token provided, then you cannot get even an empty json as a response. A JWT token must be given and verified.
Failure	403 Forbidden	The name must be unique.
Failure	404 Not Found	No boat with this boat_id exists.
Failure	406 Not Acceptable	The request type does not match Accept application/json
Failure	415 Unsupported Media Type	The request has the wrong content-type.

#### Response Examples

##### Success

```
Status: 303
'Location Header Updated'
```

##### Failure

```
Status: 400 Bad Request
One of the attributes does not meet the required parameters:
{"Error": "One of the attributes are in the incorrect type"}
```

One of the attributes is missing:  
{ "Error": "You must send at least all new attribute to update" }

*Failure*

Status: 401 Unauthorized

No JWT sent:

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
  <meta charset="utf-8">
  <title>Error</title>
</head>
```

```
<body>
  <pre>Unauthorized</pre>
</body>
```

JWT does not match the owner:

```
{ "Error": "No token found, or the token provided did not give you clearance to perform this action." }
```

*Failure*

Status: 403 Forbidden

```
{
  "Error": "The name of the boat must be unique."
}
```

*Failure*

Status: 404 Not Found

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

*Failure*

Status: 406 Not Acceptable

```
{
  "Error": "Not an acceptable request type."
}
```

*Failure*

Status: 415 Unsupported Media Type

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

}



## Delete a Boat (Protected)

Allows you to delete a boat. Note that if the boat has a load, deleting the boat makes the loads carrier attribute empty.

DELETE /boats/:boat\_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request

Request Body

None

Authorization Header

Bearer JWT

Response

Response Body

Required

Response Body Format

None

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	IF the JWT sub does not match the owner variable of the boat. Then this action cannot be done.
Failure	404 Not Found	No boat with this boat_id exists

Response Examples

*Success*

Status: 204 No Content

*Failure*

Status: 401 Not Found

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

*Failure*

Status: 404 Not Found

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

## Create a Load (Unprotected)

Allows you to create a new slip.

POST /loads

Request

Request Body

JSON

Authorization Header

None

Response

Response Body

Required

Response Body Format

JSON

Request JSON Attributes

Name	Description	Required?
Volume	The volume of the load	Yes
Item	The item in the load	Yes
creation_date	The date the load is created	Yes

Request Body Example

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

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	<p>If the request is missing the number attribute, the slip must not be created, and 400 status code must be returned.</p> <p>You don't need to validate the values of this attribute and can assume that if the number attribute is specified, then its value is valid.</p> <p>You can also assume that the request will not contain any extraneous attribute (i.e., the request JSON will never</p>

		contain any attribute other than number).
Failure	406 Not Acceptable	The request type does not match Accept application/json
Failure	415 Unsupported Media Type	The request has the wrong content-type.

#### 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 `current_boat` is the ID of the boat currently at this slip. If there is no boat at this slip, the value of `current_boat` should be null.

#### Success

Status: 201 Created

```
{
  "id": 123,
  "number": 1,
  "current_boat": null
}
```

#### Failure

Status: 400 Bad Request

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

#### Failure

Status: 406 Not Acceptable

```
{
  "Error": " Not an acceptable request type."
}
```

#### Failure

Status: 415 Unsupported Media Type

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

## Get a Load (Unprotected)

Allows you to get an existing load.

GET /loads/:load\_id

Request

Path Parameters

Name	Description
load_id	ID of the load

Request

Request Body

None

Authorization Header

None

Response

Response Body

Required

Request Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No slip with this slip_id exists
Failure	406 Not Acceptable	The wrong accept request type was used

Response Examples

*Success*

Status: 200 OK

```
{
  "id": 123,
  "number": 1
  "current_boat": 345
}
```

*Failure*

Status: 404 Not Found

```
{
  "Error": "No slip with this slip_id exists"
}
```

*Failure*

Status: 406 Not Acceptable

```
{
  "Error": " Not an acceptable request type."
}
```

## List all Loads (Unprotected)

List all the loads.

GET /loafds

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 wrong accept request type was used

Response Examples

*Success*

Status: 200 OK

```
[
  {
    "id": 123,
    "number": 1,
    "current_boat": 345
  },
  {
    "id": 456,
    "number": 2,
    "current_boat": null
  }
]
```

*Failure*

Status: 406 Not Acceptable

```
{
  "Error": " Not an acceptable request type."
}
```

## Delete a Load (Unprotected)

Allows you to delete a Load. If the load being deleted has a boat, the boat will lose the load in its load attribute.

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 slip with this slip_id exists

### Response Examples

#### Success

Status: 204 No Content

#### Failure

Status: 404 Not Found

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

## Boat receives a load (Protected)

Boat receives a load.

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 Body

None

### Authorization Header

Bearer JWT

### 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 empty.
Failure	401 Unauthorized	The JWT token does not match the boats owner, or no JWT was provided at all.
Failure	403 Forbidden	The load is not empty because there is already a boat with this load.
Failure	404 Not Found	The specified boat and/or load does not exist
Failure	406 Not Acceptable	Wrong accept request type.

### Response Examples

#### Success

Status: 204 No Content

#### Failure

Status: 401 Unauthorized

```
{
  "Error": " No token found, or the token provided did not give you clearance to
perform this action."
}
```



#### *Failure*

Status: 403 Forbidden

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

#### *Failure*

Status: 404 Not Found

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

#### *Failure*

Status: 406 Not Acceptable

```
{  
"Error": "Not an acceptable request type."  
}
```

#### *Comment*

- A boat can have multiple loads.
- Loads can only have 1 carrier
- If there is a load/boat connection on the event that one of these entities gets removed from the datastore, the remaining entities attribute will reflect that.

## Boat removes a Load (Protected)

Boat has left the slip to go to sea. The slip is now empty.

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 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 slip exists with this slip_id and this boat is at this slip.
Failure	401 Unauthorized	The JWT token (if it was provided) is not the owner of the boat.
Failure	403 Forbidden	The load/boat connection is not correct. The load is most likely assigned to another boat or didn't assigned to that boat at all.
Failure	404 Not Found	No boat with this boat_id is at the slip with this slip_id. This could be because no boat with this boat_id exists, or because no slip with slip_id exists, or even if both boat_id and slip_id are valid, the boat with this boat_id is not at this slip with this slip_id

#### Response Examples

##### Success

Status: 204 No Content

##### Failure

Status: 401 Unauthorized

```
{
  "Error": "No token found, or the token provided did not give you clearance to perform this action."
}
```

##### Failure

Status: 403 Forbidden

```
{
```

```
"Error": " The load isnt assigned to this boat"  
}
```

#### *Failure*

```
Status: 404 Not Found  
{  
"Error": " No boat with this boat_id is loaded with the load with this load_id"  
}
```