

# Final Project

CS 493: Cloud Application Development

Fall 2023

Sefath Khan

My URL .....	1
Data Model .....	2
Create a Boat .....	3
View a Boat .....	6
View all Boats for User .....	8
Edit a Boat - PUT .....	11
Edit a Boat - PATCH .....	14
Delete a Boat.....	16
Create a Load .....	18
View a Load .....	20
View all Loads.....	21
Edit a Load - PUT .....	23
Edit a Load - PATCH.....	25
Delete a Load .....	27
Assign a Load to a Boat .....	28
Remove a Load from a Boat.....	29
View all Users.....	30

## My URL

Type	URL	Date
Where application is deployed and for account creation/login	<a href="https://khanse-project.wn.r.appspot.com">https://khanse-project.wn.r.appspot.com</a>	December 11, 2023

## Data Model

The app stores three kinds of entities in Datastore, Users and two non-user entities: Boats and Loads.

### Users

Users are a user entity that has 1 protected entity called Boats. Users can be viewed through an unprotected endpoint but can solely do CRUD operations on boats and assign or remove loads from their boats.

Property	Required?	Data Type	Notes
id	Datastore generates it	Integer	The id of the user. Datastore automatically generates it. Don't add it yourself as a property of the entity.
user_name	yes	String	User's email from Auth0
user_id	yes	String	User's id to identify who owns the boat. It is the 'sub' property of the 'userinfo' property from the jwt obtained when a user creates an account. Example: 'auth0 8o34naq8765efc8jl89df92p'
self	Self generated	String	Self-link that goes to the most direct location to find the user. Cannot store in Datastore, must be generated each time.

### Boats

Boats is a protected non-user entity that is only accessible through corresponding JWT provided by client. The JWT is generated through the Auth0 process when first accessing the URL.

Property	Data Type	Required?	Notes
id	Integer	Datastore generates it	The id of the boat. Datastore automatically generates it. Don't add it yourself as a property of the entity.
name	String	yes	Name of the boat.
type	String	yes	Type of boat. E.g., Sailboat, Catamaran, etc.
length	Integer	yes	The length of the boat in feet.
loads	List	no	Loads of the boat, embedded list. Each load has ID and self-URL. Can be empty when creating boat.
owner	string	yes	Boats can only be created, deleted, read, and updated by the associated user and jwt. User's id will be updated here when each boat is created.
self	String	Self generated	Self-link that goes to the most direct location to find the boat. Cannot store in Datastore, must be generated each time.

### Loads

Loads is an unprotected non-user entity that can be edited by anyone. However if the load is on a protected resource like a boat, then assigning or removing the load to the specific boat can only be done by the permission of the corresponding JWT token.

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.
carrier	Dictionary	The boat carrying the load. ID and self-URL. Can be 'null' at creation.
item	String	Item type of load like Toys or Shoes.
creation_date	String	Date the load was created.
self	String	Self-link that goes to the most direct location to find the load. Cannot store in Datastore, must be generated each time.

## Create a Boat

Allows you to create a new boat.

POST /boats

### Request

#### Path Parameters

None

#### Content-Type Header

Required: application/json

#### Authorization Header

Required: Bearer Token (JWT)

#### Accept Header

Required: application/json or \*/\*

#### Request Body

Required

#### Request Body Format

JSON

#### Request JSON Attributes

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

#### Request Body Example

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

## Response

### Response Body Format

#### JSON

### Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	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	Missing, invalid JWT. Boat cannot be created without valid JWT.
Failure	406 Not Acceptable	Unacceptable mimetype sent by client to the server. Request Accept header does not contain 'application/json'

### Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. Your app must send back this value in the response body as shown in the example.

#### Success

Status: 201 Created

```
{
  "type": "Catamaran",
  "loads": [],
  "length": 28,
  "name": "Sea Witch",
  "id": 123,
  "owner": "auth0|123456789123456789"
  "self": "http://appspot.com/boats/123"
}
```

#### Failure

Status: 400 Bad Request

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

#### Failure

Status: 401 Unauthorized

```
{
  "code": "no auth header",
  "description": "Authorization header is missing"
}
```

### *Failure*

Status: 406 Not Acceptable

```
{  
  "Error": "Not Acceptable"  
}
```

## View a Boat

Allows you to get an existing boat

GET /boats/:boat_id
---------------------

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Body

None

Authorization Header

Required: Bearer Token (JWT)

Accept Header

Required: application/json or \*/\*

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	Missing or invalid JWT
Failure	403 Forbidden	Wrong user is trying to view someone else's boats
Failure	406 Not Acceptable	Unacceptable Accept header. Not in 'application/json'

Response Examples

*Success*

<p>Status: 200 OK</p> <pre>{   "type": "Catamaran",   "loads": [],   "length": 28,   "name": "Sea Witch",   "id": 123,   "self": "http://appspot.com/boats/123" }</pre>
---

*Failure*

<p>Status: 404 Not Found</p> <pre>{</pre>
---

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

*Failure*

Status: 401 Unauthorized

```
{
  "code": "no auth header",
  "description": "Authorization header is missing"
}
```

*Failure*

Status: 406 Not Acceptable

```
{
  "Error": "Not Acceptable"
}
```

*Failure*

Status: 403 Forbidden

```
{
  "Error": "Boat owned by another user. You are not permitted to access this resource."
}
```

## View all Boats for User

List all the boats.

GET /boats

Request

Path Parameters

None

Authorization Header

Required: Bearer Token (JWT)

Accept Header

Required: application/json or \*/\*

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Must have self-URLs and next link generate by itself
Failure	406 Not Acceptable	Unaccepted mimetype sent by client to the server. Must be 'application/json'
Failure	401 Unauthorized	Missing or invalid JWT
Failure	403 Forbidden	Wrong user viewing someone else's boats

Response Examples

Success

Status: 200 OK

```
{
  "boats": [
    {
      "id": 4797943718608896,
      "length": 29,
      "loads": [],
      "name": "Hera",
      "owner": "auth0|6577ea615eb81f7347a376ea",
      "self": "https://appspot.com/boats/4797943718608896",
      "type": "Trawler"
    },
    {
      "id": 5637476211228672,
      "length": 87,
      "loads": [],
```



```

        "name": "Poseidon",
        "owner": "auth0|6577ea615eb81f7347a376ea",
        "self": " https://appspot.com/boats/5637476211228672",
        "type": "Fishing Boat"
    },
    {
        "id": 5716561121771520,
        "length": 129,
        "loads": [],
        "name": "Demeter",
        "owner": "auth0|6577ea615eb81f7347a376ea",
        "self": " https://appspot.com/boats/5716561121771520",
        "type": "Mega Yacht"
    },
    {
        "id": 5923843625451520,
        "length": 289,
        "loads": [],
        "name": "Dionysus",
        "owner": "auth0|6577ea615eb81f7347a376ea",
        "self": " https://appspot.com/boats/5923843625451520",
        "type": "Cabin Cruiser"
    },
    {
        "id": 6205318602162176,
        "length": 13,
        "loads": [],
        "name": "Hades",
        "owner": "auth0|6577ea615eb81f7347a376ea",
        "self": " https://appspot.com/boats/6205318602162176",
        "type": "Pontoon"
    }
],
"count": 6,
"next": " https://appspot.com/boats?limit=5&offset=5"
}

```

#### Failure

Status: 401 Unauthorized

```

{
  "code": "no auth header",
  "description": "Authorization header is missing"
}

```

#### Failure

Status: 406 Not Acceptable

```
{  
  "Error": "Not Acceptable"  
}
```

#### *Failure*

Status: 403 Forbidden

```
{  
  "Error": "Boat owned by another user. You are not permitted to access this resource."  
}
```

## Edit a Boat - PUT

Allows you to edit a boat. Updates all fields.

PUT /boats/:boat_id
---------------------

### Request

#### Path Parameters

Name	Description
boat_id	ID of the boat

#### Request Body

Required

#### Content-Type Header

Required: application/json

#### Authorization Header

Required: Bearer Token (JWT)

#### Accept Header

Required: application/json or \*/\*

#### 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": "Ulysses",
  "type": "Sailboat",
  "length": 45
}
```

### Response

No body

#### Response Body Format

JSON

Failure: JSON

### Response

No body

#### Response Body Format

JSON

Failure: JSON

## Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Deleting a boat will unload any loads that were loaded on to it
Failure	401 Unauthorized	Missing or invalid JWT
Failure	404 Not Found	No boat with this boat_id exists
Failure	406 Not Acceptable	Returned when client wants a media type that is unsupported by the server. Read through the 'Accept' header.
Failure	405 Method Not Allowed	Returned when a client tries to make a POST request for this specific endpoint. Client cannot try to add a boat using this endpoint. Should return 'Allow' header with correct request method names.
Failure	403 Forbidden	Unauthorized user trying to edit someone else's boat

## Response Examples

### Success

Status: 204 No Content

### Failure

Status: 403 Forbidden

```
{
  "Error": "Boat owned by another user. You are not permitted to delete this boat"
}
```

### Failure

Status: 404 Not Found

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

### Failure

Status: 406 Not Acceptable

```
{
  "Error": "Not Acceptable"
}
```

### Failure

Status: 405 Method Not Allowed

```
{
  "Error": "Method Not Allowed"
}
```

### *Failure*

Status: 401 Unauthorized

```
{  
  "code": "no auth header",  
  "description": "Authorization header is missing"  
}
```

## Edit a Boat - PATCH

Allows you to edit a boat. Updates only edited fields.

PUT /boats/:boat_id
---------------------

### Request

#### Path Parameters

Name	Description
boat_id	ID of the boat

#### Request Body

Required

#### Content-Type Header

Required: application/json

#### Authorization Header

Required: Bearer Token (JWT)

#### Accept Header

Required: application/json or \*/\*

#### Request JSON Attributes

Name	Description	Required?
name	The name of the boat.	No
type	The type of the boat. E.g., Sailboat, Catamaran, etc.	No
length	Length of the boat in feet.	No

#### Request Body Example

```
{
  "name": "Hercules",
}
```

### Response

No body

#### Response Body Format

JSON

Failure: JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Deleting a boat will unload any loads that were loaded on to it
Failure	401 Unauthorized	Missing or invalid JWT
Failure	404 Not Found	No boat with this boat_id exists

Failure	406 Not Acceptable	Returned when client wants a media type that is unsupported by the server. Read through the 'Accept' header.
Failure	405 Method Not Allowed	Returned when a client tries to make a POST request for this specific endpoint. Client cannot try to add a boat using this endpoint. Should return 'Allow' header with correct request method names.
Failure	403 Forbidden	Unauthorized user trying to edit someone else's boat

## Response Examples

### Success

Status: 204 No Content
------------------------

### Failure

<b>Status: 403 Forbidden</b> <pre>{   "Error": "Boat owned by another user. You are not permitted to delete this boat" }</pre>
---

### Failure

<b>Status: 404 Not Found</b> <pre>{   "Error": "No boat with this boat_id exists" }</pre>
--

### Failure

<b>Status: 406 Not Acceptable</b> <pre>{   "Error": "Not Acceptable" }</pre>
---

### Failure

<b>Status: 405 Method Not Allowed</b> <pre>{   "Error": "Method Not Allowed" }</pre>
---

### Failure

<b>Status: 401 Unauthorized</b> <pre>{   "code": "no auth header",   "description": "Authorization header is missing" }</pre>
--

## Delete a Boat

Allows you to delete a boat. Note: if a load was on this boat, the load will be reassigned to have no carrier when boat is deleted.

DELETE /boats/:boat_id
------------------------

### Request

#### Path Parameters

Name	Description
boat_id	ID of the boat

### Request Body

None

### Authorization Header

Required: Bearer Token (JWT)

### Response

No body

### Response Body Format

Success: No body

Failure: JSON

### Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Deleting a boat will unload any loads that were loaded on to it
Failure	404 Not Found	No boat with this boat_id exists
Failure	403 Forbidden	Unauthorized user trying to delete someone else's boat
Failure	405 Method Not Allowed	Returned when a client tries to make a POST request for this specific endpoint. Client cannot try to add a boat using this endpoint. Should return 'Allow' header with correct request method names.

### Response Examples

#### Success

Status: 204 No Content
------------------------

#### Failure

Status: 404 Not Found { "Error": "No boat with this boat_id exists" }
--

#### Failure

Status: 405 Method Not Allowed { "Error": "Method Not Allowed" }
---



*Failure*

Status: 403 Forbidden

```
{  
  "Error": "Boat owned by another user. You are not permitted to delete this boat"  
}
```

## Create a Load

Allows you to create a new load.

POST /loads
-------------

Request

Path Parameters

None

Content-Type Header

Required: application/json

Accept Header

Required: application/json or \*/\*

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
volume	The volume of the load, an integer.	Yes
item	Item of the load	Yes
creation_date	Date the load was created	Yes

Request Body Example

```
{
  "volume": 10,
  "item": "Stuffed Animals",
  "creation_date": "10/19/2021"
}
```

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 load 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>

		You can also assume that the request will not contain any extraneous attribute (i.e., the request JSON will never contain any attribute other than number).
Failure	401 Unauthorized	Missing, invalid JWT. Boat cannot be created without valid JWT.
Failure	406 Not Acceptable	Unacceptable mimetype sent by client to the server. Request Accept header does not contain 'application/json'

### 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 `carrier` is the ID of the boat currently holding this load. If there is no boat holding this load, the value of `carrier` should be null.
- Self-URL should be generated not stored in the Datastore.
- Is not tied to any users

### Success

Status: 201 Created

```
{
  "volume": 10,
  "creation_date": "10/19/2021",
  "carrier": null,
  "item": "Stuffed Animals",
  "id": 123,
  "self": "http://appspot.com/loads/123"
}
```

### Failure

Status: 400 Bad Request

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

## View a Load

Allows you to get an existing load.

GET /loads/: load_id
----------------------

### Request

#### Path Parameters

Name	Description
load_id	ID of the load

### Request Body

None

### Response

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No load with this load_id exists.

#### Response Examples

##### Success

Status: 200 OK
<pre>{   "volume": 5,   "creation_date": "10/18/2021",   "carrier": null,   "item": "LEGO Blocks",   "id": 456,   "self": "http://appspot.com/loads/456" }</pre>

##### Failure

Status: 404 Not Found
<pre>{"Error": "No load with this load_id exists"}</pre>

## View all Loads

List all the loads.

GET / loads
-------------

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Must have self-URLs and next link generate by itself

Response Examples

*Success*

Status: 200 OK

```
{
  "loads": [
    {
      "volume": 10,
      "creation_date": "10/19/2021",
      "carrier": {
        "id": 123,
        "self": "http://appspot.com/boats/123"
      },
      "item": "Stuffed Animals",
      "id": 321,
      "self": "http://appspot.com/loads/321"
    },
    {
      "volume": 3,
      "creation_date": "10/19/2021",
      "carrier": {
        "id": 123,
        "self": "http://appspot.com/boats/123"
      },
      "item": "Food",
      "id": 654,
      "self": "http://appspot.com/loads/654"
    },
    {

```

```
        "volume": 7,  
        "creation_date": "10/20/2021",  
        "carrier": {  
            "id": 123,  
            "self": "http://appspot.com/boats/123"  
        },  
        "item": "Furnitures",  
        "id": 654,  
        "self": "http://appspot.com/loads/654"  
    }  
],  
    "next": "http://appspot.com/loads/?limit=3&offset=3"  
}
```

#### *Failure*

**Status: 401 Unauthorized**

```
{  
    "code": "no auth header",  
    "description": "Authorization header is missing"  
}
```

#### *Failure*

**Status: 406 Not Acceptable**

```
{  
    "Error": "Not Acceptable"  
}
```

## Edit a Load - PUT

Allows you to edit a load. Updates all fields.

PUT /loads/:load_id
---------------------

### Request

#### Path Parameters

Name	Description
load_id	ID of the load

#### Request Body

Required

#### Content-Type Header

Required: application/json

#### Authorization Header

Required: Bearer Token (JWT)

#### Accept Header

Required: application/json or \*/\*

#### Request JSON Attributes

Name	Description	Required?
volume	The name of the load.	Yes
Creation_date	When the load was created.	Yes
Item	Type of load	Yes

#### Request Body Example

```
{
  "volume": 10,
  "item": "Stuffed Animals",
  "creation_date": "10/19/2021"
}
```

### Response

No body

#### Response Body Format

JSON

Failure: JSON

### Response

No body

#### Response Body Format

JSON

Failure: JSON

## Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Deleting a boat will unload any loads that were loaded on to it
Failure	401 Unauthorized	Missing or invalid JWT
Failure	404 Not Found	No boat with this boat_id exists
Failure	406 Not Acceptable	Returned when client wants a media type that is unsupported by the server. Read through the 'Accept' header.
Failure	405 Method Not Allowed	Returned when a client tries to make a POST request for this specific endpoint. Client cannot try to add a boat using this endpoint. Should return 'Allow' header with correct request method names.

## Response Examples

### Success

Status: 204 No Content

### Failure

Status: 404 Not Found

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

### Failure

Status: 406 Not Acceptable

```
{
  "Error": "Not Acceptable"
}
```

### Failure

Status: 405 Method Not Allowed

```
{
  "Error": "Method Not Allowed"
}
```



## Edit a Load - PATCH

Allows you to edit a load. Updates only edited fields.

PUT /loads/:load_id
---------------------

### Request

#### Path Parameters

Name	Description
boat_id	ID of the boat

#### Request Body

Required

#### Content-Type Header

Required: application/json

#### Authorization Header

Required: Bearer Token (JWT)

#### Accept Header

Required: application/json or \*/\*

#### Request JSON Attributes

Name	Description	Required?
volume	The name of the load.	Yes
Creation_date	When the load was created.	Yes
Item	Type of load	Yes

#### Request Body Example

```
{
  "item": "Eggplants",
}
```

### Response

No body

#### Response Body Format

JSON

Failure: JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Deleting a load will unload any loads that were loaded on a boat
Failure	401 Unauthorized	Missing or invalid JWT
Failure	404 Not Found	No load with this load_id exists

Failure	406 Not Acceptable	Returned when client wants a media type that is unsupported by the server. Read through the 'Accept' header.
Failure	405 Method Not Allowed	Returned when a client tries to make a POST request for this specific endpoint. Client cannot try to add a boat using this endpoint. Should return 'Allow' header with correct request method names.

## Response Examples

### Success

Status: 204 No Content
------------------------

### Failure

Status: 404 Not Found <pre>{   "Error": " No boat with this load_id exists" }</pre>
--

### Failure

Status: 406 Not Acceptable <pre>{   "Error": "Not Acceptable" }</pre>
--

### Failure

Status: 405 Method Not Allowed <pre>{   "Error": "Method Not Allowed" }</pre>
--

## Delete a Load

Allows you to delete a load. If the load being deleted has a boat carrying it, the boat will also lose the load.

DELETE / loads/: load_id
--------------------------

### Request

#### Path Parameters

Name	Description
load_id	ID of the load

### Request Body

None

### Response

No body

#### Response Body Format

Success: No body

Failure: JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	404 Not Found	No load with this load_id exists

### Response Examples

#### Success

Status: 204 No Content
------------------------

#### Failure

Status: 404 Not Found  <pre>{   "Error": "No load with this load_id exists" }</pre>
---

## Assign a Load to a Boat

Load is assigned to boat

```
PUT /boats/{{boat_id}}/loads/{{load_id}}
```

### Request

#### Path Parameters

Name	Description
load_id	ID of the load
boat_id	ID of the boat

### Request Body

None

### Response

No body

#### Response Body Format

Success: No body

Failure: JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	Succeeds only if a boat exists with this boat_id, a load exists with load_id and load does not have a carrier.
Failure	403 Forbidden	The load is already loaded on another boat
Failure	404 Not Found	The specified boat and/or load does not exist

#### Response Examples

##### Success

```
Status: 204 No Content
```

##### Failure

```
Status: 403 Forbidden

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

```
Status: 404 Not Found

{"Error": "No resource with either load_id or boat_id exists."}
```

## Remove a Load from a Boat

Boat should have load deleted from 'loads' list and load deleted from all loads

```
DELETE /boats/{boat_id}/loads/{load_id}
```

### Request

#### Path Parameters

Name	Description
load_id	ID of the load
boat_id	ID of the boat

### Request Body

None

### Response

No body

#### Response Body Format

Success: No body

Failure: JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	404 Not Found	No boat or load was found with these ids
Failure	403 Forbidden	Wrong user is trying to access boat's loads

#### Response Examples

##### Success

```
Status: 204 No Content
```

##### Failure

```
Status: 404 Not Found
"Error": "No resource with either load_id or boat_id exists."
```

##### Failure

```
Status: 403 Forbidden
{
  "Error": "Boat owned by another user. You are not permitted to access this resource."
}
```

## View all Users

List all the users.

GET /users

Request

Path Parameters

None

Accept Header

Required: application/json or \*/\*

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	Must have self-URLs and next link generate by itself
Failure	406 Not Acceptable	Request Accept header does not accept 'application/json'

Response Examples

Success

Status: 200 OK

```
[
  {
    "id": 5072058866204672,
    "self": "http://appspot.com/users/5072058866204672",
    "user_id": "auth0|657794c9a502150ac548a198",
    "user_name": "hamara@cookie.com"
  },
  {
    "id": 5080330100801536,
    "self": "http://appspot.com/users/5080330100801536",
    "user_id": "auth0|6577ace85a04ce45f9a7b293",
    "user_name": "khushi@bread.com"
  },
]
```

### *Failure*

Status: 406 Not Acceptable

```
{  
  "Error": "Not Acceptable"  
}
```