

CS493 Portfolio Assignment: Final Project

By: Elif SEKER

The URL where your application is deployed on GCP:

<https://cs493portfolio-389502.wl.r.appspot.com>

The URL for account creation/login:

<https://cs493portfolio-389502.wl.r.appspot.com>

A data model:

The app stores two kinds of non-user entities in Datastore, “Boats” and “Loads”.

Also it has “Owner” entity to model the user.

Boats has loads and owner property.

Loads has a carrier property.

All of the boats related endpoints are protected with JWT because of owner relationship.

Boats

Property	Data Type	Notes
id	integer	The id of the boat. Datastore automatically generates it.
name	string	Name of the boat.
type	string	The type of boat, power boat, sailboat, catamaran etc.
length	integer	The length of the boat.
public	boolean	true if the boat is public, false if it's private.
owner	string	The boat's owner, value of sub property in the JWT
loads	dictionary	Details in Loads entity

self	string	The URL link of resource
------	--------	--------------------------

Loads

Property	Data Type	Notes
id	integer	The id of the load. Datastore automatically generates it.
volume	integer	The volume of the load.
carrier	dictionary	The data of Boats entity's id,name and self.
item	string	The name of the load.
creation_date	string	Date the load was created.
self	string	The URL link of resource

Owners

Pre-created 2 users info:

- User1:
 - Email: beautiful@day.com
 - Password: Watermelon123
- User2:
 - Email: lovely@day.com
 - Password: Yesil123

The API specifications:

Create a boat

Allows you to create a new boat.

POST /boats

Request

Path Parameters

None

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.

Response Examples

Success

```
Status: 201 Created

{
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "loads": [],
  "self": "https://appspot.com/boats/1234"
}
```

Failure

```
Status: 400 Bad Request

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

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

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

Response Examples

Success

```
Status: 200 OK
{
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "loads": [],
  "self": "https://appspot.com/boats/1234"
}
```

Failure

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

View all boats

List all the boats.

```
GET /boats
```

Request

Path 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
{"boats":
  [{"type": "Yacht",
    "loads":
      [{"id": 6379915934957568, "self": "http://127.0.0.1:8080/loads/6379915934957568"},
        {"id": 5661929137766400, "self": "http://127.0.0.1:8080/loads/5661929137766400"}, {"id":
          5414233810403328, "self":
            "http://127.0.0.1:8080/loads/5414233810403328"}, {"id": 6635205435588608, "self":
              "http://127.0.0.1:8080/loads/6635205435588608"}]},
    "length": 15,
    "id": 5040462201094144,
    "self": null,
    "name": "Lucifer"},
  {"type": "Sailing Boat",
    "loads": [],
    "length": 13,
    "self": null,
    "name": "Calypso",
    "id": 5107674546962432},
  {"type": "Motor Boat",
    "loads": [],
    "length": 25,
    "self": null,
    "name": "Abraham",
    "id": 5670624500383744}},
  "next": "http://127.0.0.1:8080/boats?limit=3&offset=3"}
```

Delete a boat

Allows you to delete a boat. Unload any loads that were loaded on to it.

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

Request

Path Parameters

Name	Description
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 with this boat_id exists
---------	---------------	----------------------------------

Response Examples

Success

Status: 204 No Content

Failure

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

Edit a boat

Allows you to edit a boat.

PUT

Allows updating all attributes are modified.

PUT /boats/:boat_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Body

Required

Request Body Format

Request must be 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

Response must be JSON.

Response Statuses

Outcome	Status Code	Notes
Success	303 See Other	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, the boat 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": 123,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "self": "https://appspot.com/boats/123"
}
```

Failure

```
Status: 400 Bad Request
{
  "Error": "The request object is missing at least one of the required attributes"
}
```

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

PATCH

Allows updating any subset of attributes while the other attributes remain unchanged.

```
PATCH /boats/:boat_id
```

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Body

Required

Request Body Format

Request must be JSON.

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 Examples

```
{
  "name": "Sea Witch",
}
```

```
{
  "id": 345678
}
```

Response

Response Body Format

Response must be JSON.

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad request	Updating the id is not allowed
Failure	404 Not Found	No boat with this boat_id exists

Response Examples

Success

```
Status: 200 OK
{
  "id": 123,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "self": "https://appspot.com/boats/123"
}
```

Failure

```
Status: 400 Bad request
{
  "Error": "Updating the id is not allowed"
}
```

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


Create a load

Allows you to create a new load.

POST /loads

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
volume	The volume of the load.	Yes
item	The name of the load.	Yes
creation_date	Date the load was 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	If the request is missing any of the 3 required attributes, the load must not be created, and 400 status code must be returned.

Response Examples

Success

Status: 201 Created

```
{
  "volume": 5,
  "creation_date": "10/18/2021",
  "carrier": null,
  "item": "LEGO Blocks",
  "self": "http://127.0.0.1:8080/loads/5540819314409472",
  "id": 5540819314409472
}
```

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

```
{
  "volume": 5,
  "creation_date": "10/18/2021",
  "carrier": null,
  "item": "LEGO Blocks",
  "self": "http://127.0.0.1:8080/loads/5540819314409472",
  "id": 5540819314409472
}
```

Failure

Status: 404 Not Found

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

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	

Response Examples

Success

Status: 200 OK

```
{
  "loads": [
    {
      "volume": 7,
      "creation_date": "10/20/2021",
      "carrier": {
        "id": 5040462201094144,
        "self": "http://127.0.0.1:8080/boats/5040462201094144",
        "item": "Furnitures",
        "id": 5414233810403328,
        "self": "http://127.0.0.1:8080/loads/5414233810403328"
      },
      "volume": 3,
      "creation_date": "10/19/2021",
      "carrier": {
        "id": 5040462201094144,
        "self": "http://127.0.0.1:8080/boats/5040462201094144",
        "item": "Food",
        "id": 5661929137766400,
        "self": "http://127.0.0.1:8080/loads/5661929137766400"
      },
      "volume": 10,
      "creation_date": "10/19/2021",
      "carrier": {
        "id": 5040462201094144,
        "self": "http://127.0.0.1:8080/boats/5040462201094144",
        "item": "Stuffed Animals",
        "id": 6379915934957568,
        "self": "http://127.0.0.1:8080/loads/6379915934957568"
      }
    ]
  },
  "next": "http://127.0.0.1:8080/loads?limit=3&offset=3"
}
```

Delete a load

Allows you to delete a load. Update the boat that was carrying it.

```
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

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

Edit a load

Allows you to edit a boat.

PUT

Allows updating all attributes are modified.

PUT /boats/:boat_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Body

Required

Request Body Format

Request must be 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

Response must be JSON.

Response Statuses

Outcome	Status Code	Notes
Success	303 See Other	

Failure	400 Bad Request	If the request is missing any of the 3 required attributes, the boat 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": 123,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "self": "https://appspot.com/boats/123"
}
```

Failure

```
Status: 400 Bad Request
{
  "Error": "The request object is missing at least one of the required attributes"
}
```

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

PATCH

Allows updating any subset of attributes while the other attributes remain unchanged.

```
PATCH /boats/:boat_id
```

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Body

Required

Request Body Format

Request must be JSON.

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 Examples

```
{
  "name": "Sea Witch",
}
```

```
{
  "id": 345678
}
```

Response

Response Body Format

Response must be JSON.

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad request	Updating the id is not allowed
Failure	404 Not Found	No boat with this boat_id exists

Response Examples

Success

```
Status: 200 OK
{
  "id": 123,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "self": "https://appspot.com/boats/123"
}
```

Failure

```
Status: 400 Bad request
{
  "Error" : "Updating the id is not allowed"
}
```

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

Managing loads:

Assign a load to a 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

Note: Set Content-Length to 0 in your request when calling out to this endpoint.

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's carrier is null.
Failure	403 Forbidden	The slip is not empty because there is already a boat at this slip.
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": "The specified boat and/or load does not exist"
}
```

Remove a load from a boat without deleting that load

```
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	Succeeds only if a boat exists with this boat_id, a load exists with this load_id and this load is at this boat.
Failure	404 Not Found	No boat with this boat_id is loaded with the load with this load_id. This could be because no boat with this boat_id exists, or because no load with load_id exists, or even if both boat_id and load_id are valid, the load with this load_id is not at this load with this load_id.

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found

```
{
  "Error": "No boat with this boat_id is loaded with the load with this load_id"
}
```

Create an owner

Allows you to create a new owner.

POST /owners

Request

Path Parameters

None

Request Body

Required

Request Body Format

Request must be JSON.

Request JSON Attributes

Name	Description	Required?
owner_id	The unique id of the owner.	Yes

Request Body Example

```
{
  "owner_id": "auth0|646d33e34467e7166eb935b1"
}
```

Response

Response Body Format

Response must be JSON.

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	

Response Examples

Success

Status: 201 Created

```
{
  "owner_id": "auth0|646d33e34467e7166eb935b1"
}
```

Get the owner's boats

List all boats of the owner.

```
GET /owners/<owner_id>/boats
```

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

Response must be JSON.

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	

Response Examples

```
"boats": [
  {
    "name": "nope",
    "public": false,
    "loads": [],
    "length": 28,
    "type": "yat",
    "owner": "auth0|64667371d70fc3a64b089efb",
    "id": 5068721408180224
  },
  {
    "name": "sure",
    "public": false,
    "loads": [],
    "length": 28,
    "type": "yat",
    "owner": "auth0|64667371d70fc3a64b089efb",
    "id": 5080330100801536
  }
]
```

Get all owners

List all the owners.

GET /owners

Request

Path 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

```
[
{
  "owner_id": "auth0|646d33e34467e7166eb935b1",
  "id": 5706627130851328
},
{
  "owner_id": "auth0|646d33e34467e7166eb935b1",
  "id": 5716561121771520
},
{
  "owner_id": "auth0|64667371d70fc3a64b089efb",
  "id": 5915636647591936
}
]
```