REST API Spec of Boat, load and ownership:

Sophie Zhao

Part 1: API Spec

Change log

Version	Change	Date
1.0	Initial version.	May 30, 2022

Data Model

The app stores three kinds of entities in Datastore, Boats, loads and owners.

Boats (Protected entity)

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.
type	String	Type of the boat. E.g., Sailboat, Catamaran, etc.
length	Integer	The length of the boat in feet.
loads	Array of JSON Objects	All Loads on the boat.
self	String	The URL link of this boat.
owner	String	The boat's owner, value of sub property in the JWT.

Loads (Unprotected entity)

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	Object(a boat)	The boat carryig the load, including id, name and self link of a boat.
item	String	The item of the load.
creation_date	String	Date the load was created.
self	String	The URL link of the load.

Users

Property	Data Type	Notes
id	String	The id of the Owner, value of sub property in the JWT.
fname	String	The first name of the owner.
lname	String	The last name of the owner.

Create a Boat

Allows you to create a new boat.

POST /boats

Request

Path Parameters
None

Request Body
Required

Request Body Format
JSON

Protected
No

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",  # The name of the boat, a string
  "type": "Catamaran",   # The type of boat, power boat, sailboat, catamaran etc. a string
  "length": 28,         # The length of the boat, an integer
}
```

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 4 required attributes, the boat must not be created, and 400 status code must be returned. Return error: "The request object is missing at least one of the required attributes".

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


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


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

Protected

No

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No boat with this boat_id exists
Failure	406 Not Acceptable	Client don't accept application/json

Response Examples

Success

Status: 200 OK

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

Failure

Status: 404 Not Found

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

List all Boats

List all the boats.

GET /boats

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON, 3 boats per page. Implement pagination by using cursor method.

Protected

No

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

Response Examples

Success

Status: 200 OK


```
[
{
"id": 123,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "loads": []
},
{
  "id": 456,
  "name": "Adventure",
  "type": "Sailboat",
  "length": 50
  "loads": []
},
{
  "id": 789,
  "name": "Hocus Pocus",
  "type": "Sailboat",
  "length": 100
  "loads": []
}
]
```

Delete a Boat

Allows you to delete a boat. Deleting a boat will 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

Protected

Yes

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	404 Not Found	No boat with this boat_id exists
Failure	405 Not Acceptable	No boat id provided

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found

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

Create a load

Allows you to create a new load. All newly created loads should begin unassigned to any boat.

POST /loads

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
volumn	The volumn of the load.	Yes
item	The item of the load.	Yes
creation_ date	Date the load was created	Yes

Request Body Example

```
{
  "volume": 5,     # The volume of the load, an integer
  "item": "LEGO Blocks",
  "creation_date": "10/18/2020" ,
  }
```

Response

Response Body Format

JSON

Protected

No

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	

Failure	400 Bad Request	If the request is missing the number attribute, the load must not be created, and 400 status code must be returned. 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. 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).

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 load. If there is no boat at this load, the value of `current_boat` should be null.

Success

```
Status: 201 Created

{
"volume": 5,    # The volume of the load, an integer
 "carrier": null
 "item": "LEGO Blocks",
 "creation_date": "10/18/2020"
 "self":"https://appspot.com/loads/5678"
}
```

Failure

```
Status: 400 Bad Request

{
"Error":  "The request object is missing the required number"
}
```

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

Protected

No

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,
  "carrier": null
  "item": "LEGO Blocks",
  "creation_date": "10/18/2020"
  "self":"https://appspot.com/loads/5678"
}
```

Failure

```
Status: 404 Not Found

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

View all loads

List all the loads. Implement pagination similar to view all boats, i.e., 3 loads per page and a next link.

GET /loads

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Protected

No

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

Response Examples

Success

Status: 200 OK


```
[
{
 "volume": 5,
  "carrier": null
  "item": "LEGO Blocks",
  "creation_date": "10/18/2020"
  "self":"https://appspot.com/loads/5678"
},
{
 "volume": 5,
  "carrier": null
  "item": "LEGO Blocks",
  "creation_date": "10/18/2020"
  "self":"https://appspot.com/loads/5679"
}
]
```

Delete a load

Allows you to delete a load. Deleting a load should 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

Protected

No

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"
}
```

Assign load to boat

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

Response

No body

Response Body Format

Success: No body

Failure: JSON

Protected

No

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 assigned to a boat yet.
Failure	403 Forbidden	If a load is already assigned to one boat and then is assigned to another boat without first being removed, it should return a 403 status code.
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"
}
```

Comment

- A load cannot be assigned to multiple boats.
- However, a boat can have multiple loads.

Romove load from a boat

Remove load from the boat. The load's carrier is now null.

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

Protected

No

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 boat is at this load.
Failure	404 Not Found	No boat with this boat_id is at 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 boat with this boat_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"
}
```

View all loads for a boat

List all loads for a boat. This doesn't require pagination.

GET /boats/:boat_id/loads

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Body

None

Response

[{

"volume": 3

"item": "LEGO Blocks",

"creation_date": "10/18/2020" # Date the load was created

"self":"https://appspot.com/loads/5678"

},

{

"volume": 4

"item": "LEGO Blocks",

"creation_date": "10/18/2020" # Date the load was created

"self":"https://appspot.com/loads/5679"

},

]

Response Body Format

Success: No body

Failure: JSON

Protected

No

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 boat is at this load.
Failure	404 Not Found	No boat with this boat_id.

Response Examples

Success

```
Status: 204 No Content
```

Failure

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

Create a User

Allows you to create a new boat.

```
POST /users
```

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Protected

Yes

Request JSON Attributes

Name	Description	Required?
fname	First name of the user.	Yes
lname	Last name of the user	Yes
id	id of the user	Yes

Request Body Example

```
{
   "fname": "Summy",
   "lname": "Catty"
}
```

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 user must not be created, and 400 status code must be returned. Return error: "The request object is missing at least one of the required attributes".

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

{
   "lname": "Catty",
   "fname": "Summy",
   "id": "auth0|627d90489d042c006938cab4"
}
```

Failure

```
Status: 400 Bad Request

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

List all Users

List all the boats.

GET /users

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON, 3 boats per page. Implement pagination by using cursor method.

Protected

No

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

Response Examples

Success

Status: 200 OK


```
{
   "users": [
      {
         "lname": "Catty",
         "id": "4647336663842816",
         "fname": "Summy"
      },
      {
         "fname": "Summy",
         "id": "4793933427113984",
         "lname": "Catty"
      },
      {
         "fname": "Summy",
         "lname": "Catty",
         "id": "5151539786153984"
      },
      {
         "lname": "Catty",
         "id": "6243479118151680",
         "fname": "Summy"
      }
   ]
}
```