Assignment 4: Boat and Load Management API
API Spec

Jacob Anderson

CS 493: Cloud Application Development

Fall 2020

Oregon State University

Last Update: Oct 26, 2020. 8:00 pm Pacific

Link to API: https://anderja8-marina-load-manager.appspot.com

Change log

Page	Change	Date
	Initial version.	Oct 21, 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": "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 will not be created, and 400 status code will be returned. Any extraneous attributes in the request body will be ignored.

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.
- The `self` attribute will contain the live link to the REST resource corresponding to the new boat.
- The loads attribute is created by searching the loads datastore for loads with carrier equal to the id of the boat. For newly created boats, this will always be an empty array.

Success

```
Status: 201 Created

{
  "id": "abc123",
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "self": "https://anderja8-marina-load-manager.appspot.com/boats/abc123"
  "loads": []
}
```

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. This is also the route for viewing all loads of a given boat, via the "loads" attribute in the response body.

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

Response Examples

Success

```
Status: 200 OK

{
  "id": "abc123",
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "loads":[
     {
        "id":"123abc",
        "self": "https://anderja8-marina-load-manager.appspot.com/loads/123abc"
     },
     {
        "id:"124abc",
        "self": "https://anderja8-marina-load-manager.appspot.com/loads/124abc"
     },
  ],
  "self": "https://anderja8-marina-load-manager.appspot.com/boats/abc123"
}
```

Failure

```
Status: 404 Not Found

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

View all Boats

Allows you to view all boats. Uses pagination to display three boats per page in an array named "boats". If there are more boats, there will also be a "next" attribute in the response. Performing a GET on the URL stored in this "next" attribute will return the next page of results.

GET /boats

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
{
  "boats": [
     {
         "id": "abc123",
         "name": "Sea Witch",
     "type": "Catamaran",
         "length": 28,
     "loads":[
       {
         "id:"123abc",
         "self": "https://anderja8-marina-load-manager.appspot.com/loads/123abc"
       },
       {
         "id:"124abc",
         "self": "https://anderja8-marina-load-manager.appspot.com/loads/124abc"
       }
     ],
         "self": "https://anderja8-marina-load-manager.appspot.com/boats/abc123"
     },
     {
         "id": "abc134",
         "name": "Sea King",
     "type": "Schooner",
         "length": 15,
     "loads":[],
         "self": "https://anderja8-marina-load-manager.appspot.com/boats/abc134"
     },
  ],
  "next": https://anderja8-marina-load-
manager.appspot.com/boats?endCursor=<end_cursor>
}
```

Delete a Boat

Allows you to delete a boat. Note that if the boat is carrying loads, those loads will still exist but will not have an assigned carrier.

```
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

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.

POST /loads

Request

Request Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
weight	Integer	The weight of the load.	Yes
content	string	The content of the load	Yes
delivery_date	string	The date the load should be delivered. Must be in MM/DD/YYYY format.	Yes

Request Body Example

```
{
  "weight": 400
  "content": "hungry hippos"
  "delivery_date": "10/10/2026"
}
```

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 attributes, the slip will not be created, and 400 status code will be returned. The delivery date must be in the form "MM/DD/YYYY", otherwise, it is left to the user to ensure their input data is correct. Any extraneous attributes in the request body will be ignored.

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.
- The value of the attribute self is a live link to the REST resource corresponding to this slip.

- The carrier attribute will be unassigned for newly created loads.

Status: 201 Created

```
{
  "id": "123abc",
  "weight": 400,
  "content": "hungry hippos",
  "delivery_date": "10/10/2026"
  "carrier": null
  "self": "https://anderja8-marina-load-manager.appspot.com/slips/123abc"
}
```

Status: 400 Bad Request

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

Status: 400 Bad Request

```
{
"Error":  "The request object delivery_date could not map to a valid date. Format
must be MM/DD/YYYY"
}
```

View 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", # automatically generated by the Datastore
    "weight": 5,      #The weight of the load
    "carrier":{
      "id": abc123,
      "name": "Sea Witch",
      "self": "https://anderja8-marina-load-manager.appspot.com/boats/abc123"
    },  #The boat carrying the load
    "content":"LEGO Blocks",
    "delivery_date": "10/18/2020" #The date the load is to be delivered
    "self":"https://anderja8-marina-load-manager.appspot.com/loads/123abc"
}
```

Failure

```
Status: 404 Not Found

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

View all Loads

List all existing loads. Uses pagination to display three loads per page in an array named "loads". If there are more loads, there will also be a "next" attribute in the response. Performing a GET on the URL stored in this "next" attribute will return the next page of results.

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

{
  "loads": [
    {
      "id":"123abc",
      "weight": 5,
      "carrier":{
        "id": abc123,
        "name": "Sea Witch",
        "self": "https://anderja8-marina-load-manager.appspot.com/boats/abc123"
      },
      "content":"LEGO Blocks",
      "delivery_date": "10/18/2020"
      "self":"https://anderja8-marina-load-manager.appspot.com/loads/123abc"
    },
    {
      "id":"124abc",
      "weight": 2,
      "carrier":{
        "id": abc123,
        "name": "Sea Witch",
        "self": "https://anderja8-marina-load-manager.appspot.com/boats/abc123"
      },
      "content":"Bionicle",
      "delivery_date": "10/18/2020"
      "self":"https://anderja8-marina-load-manager.appspot.com/loads/124abc"
    }
  ],
  "next": https://anderja8-marina-load-
manager.appspot.com/boats?endCursor=<end_cursor>
}
```

Delete a Load

Allows you to delete a load. If the load being deleted has a boat, that boat will no longer show this load in its loads.

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 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"
}
```

Add a load to a boat

The load now has the boat listed as its carrier.

```
PUT /loads/:load_id/boats/:boat_id
```

Request

Request Parameters

None

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 is unassigned to any other boats.
Failure	403 Forbidden	The load is already assigned to a boat.
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": "The load is already assigned to boat with id: <boat_id>"
}
```

```
Status: 404 Not Found

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

Remove a Load from a Boat

The load is no longer has the boat as its carrier, and is now unassigned.

```
DELETE /loads/:load_id/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	Succeeds only if a boat exists with this boat_id, a load exists with this load_id and this load is assigned to this boat.
Failure	404 Not Found	No load with this load _id is assigned to a boat with this boat_id.

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 404 Not Found

{
"Error":  "No load with this load_id is assigned to a boat with this boat_id"
}
```

View all Loads for a Given Boat

Shows all loads with carrier equal to the passed boat_id. Note: this route does not use pagination.

GET /boats/:boat_id/loads

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 this load is assigned to this boat.
Failure	404 Not Found	The boat_id does not exist

Response Examples

Success

```
Status: 200 OK

[
  {
    "id":"123abc",
    "weight": 5,
    "carrier":{
      "id": abc123,
      "name": "Sea Witch",
      "self": "https://anderja8-marina-load-manager.appspot.com/boats/abc123"
    },
    "content":"LEGO Blocks",
    "delivery_date": "10/18/2020"
    "self":"https://anderja8-marina-load-manager.appspot.com/loads/123abc"
  },
  {
    "id":"124abc",
    "weight": 2,
    "carrier":{
      "id": abc123,
      "name": "Sea Witch",
      "self": "https://anderja8-marina-load-manager.appspot.com/boats/abc123"
    },
    "content":"Bionicle",
    "delivery_date": "10/18/2020"
    "self":"https://anderja8-marina-load-manager.appspot.com/loads/124abc"
  }
]
```

Failure

```
Status: 404 Not Found

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

Data Model

Two kinds of entities exist in the data model.

Boats

Name	Type	Notes
id	Integer	The id of the boat. Datastore automatically generates it.
name	String	The name of the boat
type	String	The type of the boat. E.g. sailboat, catamaran, etc.
length	Integer	The length of the boat in feet

Loads

Name	Type	Notes
id	Integer	The id of the load. Datastore automatically generates it.
weight	Integer	The weight of the load in pounds.
content	String	The content of the load.
delivery_date	String	The date the load should be delivered. Stored as a string in format MM/DD/YYYY
carrier_id	Integer	The id of the boat carrying the load. Foreign key corresponding to the boats model.