

# Assignment 9: API Spec

CS 493: Cloud Application Development

Oregon State University – Spring 2021

Terence Tang – onid: tangte

Last Update: Jun 6, 2021

Change log .....	1
Application URL.....	2
Data Model .....	2
Create a Boat .....	4
Get a Boat .....	7
List all Boats .....	9
Delete a Boat .....	11
Edit a Boat with Patch.....	13
Edit a Boat with Put .....	16
Create a Load .....	19
Get a Load .....	21
List all Loads.....	23
Delete a Load .....	25
Edit a Load with Patch .....	26
Edit a Load with Put .....	29
Assign Load to a Boat.....	31
Remove Load from a Boat .....	32
List all Users .....	33

## Change log

Version	Change	Date
1.0	Initial version.	June 4 2021

## Application URL

gcloud application URL: <https://dev-epsilon-309902.wl.r.appspot.com/marina>

application welcome / login URL: <https://dev-epsilon-309902.wl.r.appspot.com/auth/welcome>

## Data Model

In this assignment, I use Cloud Datastore to store my data. There are three kinds of entities in my data model: users, boats, and loads. All entity attributes noted below are required for each entity. Any additional attributes that may be included during entity creation will be discarded.

### *User*

Property Name	Notes
id	The id of the user. Datastore automatically generates it. Don't add it yourself as a property of the entity.
nickname	String. First name of user.
name	String. Last name of user.
email	String. User email address, e.g. user@email.com
Boats	JSON array. If user owns any boats, this array will list the id and names of their relevant boats. An owner may own many boats.
auth0_id	String. Unique identifier for user, used for relating boats to users (owners).

### **User Comments:**

- When asking for a supplied user\_id, my application expects to be handed an auth0\_id value, which it will then parse the user entities for a datastore match.
- My application parse JWT's and matches the JWT's sub value to the auth0\_id value that we store as a user's unique identifier.
- The user-to-non-user relationship I've opted for in my application is the ownership of boats by a user. A user may own many boats, and each boat can only have one owner. The value that is displayed as the owner\_id will be the auth0\_id value.

### *Boat*

Property Name	Notes
id	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	JSON array. If there are loads on the ship, then contains a list of JSON load objects which contain the id and self of load. Otherwise is an empty array. A boat many hold many loads.

owner	String. Owner unique identifier, based on user -> auth0_id value. A boat may only have one owner at a time.
-------	---

### ***Load***

Property Name	Notes
id	The id of the slip. Datastore automatically generates it. Don't add it yourself as a property of the entity.
volume	Integer. The number of the slip.
carrier	JSON. If there is a boat at the slip, then id, name and self of boat. Otherwise null. A load may only be on one boat at a time.
content	String. Description of the load contents (e.g. Lego blocks)
creation_date	String. Date the load was created in the MM/DD/YYYY format.

## Create a Boat

Allows you to create a new boat. Requires presentation of a valid JWT in order to access API endpoint to create additional boat resources.

POST /boats

### Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

### Request JSON Attributes

Name	Type	Description	Required?
name	String	The name of the boat.  Input Validation / Restrictions: <ul style="list-style-type: none"><li>Name string can only contain alphanumeric chars and spaces. e.g. a-z, A-Z, 0-9 and ' '. Any other input is invalid.</li></ul> Name string must be less than 40 characters long. Longer inputs will register as invalid.	Yes
type	String	The type of the boat. E.g., Sailboat, Catamaran, etc.  Input Validation / Restrictions: Type string must be less than 100 characters long. Longer inputs will register as invalid.	Yes
length	Integer	Length of the boat in feet.  Input Validation / Restrictions: Length value must be non-negative integer. Any trailing decimals will be truncated.	Yes

### Request Body Example

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

### Response

Response Header Example

```
{
  "Content-Type": "application/json; charset=utf-8"
}
```

## 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>Attribute input values will also be validated for correctness prior to entity creation. Extraneous attributes will be dropped and ignored during entity creation.</p>
Failure	401 Unauthorized	Protected resource requires authorization via valid bearer token in order to access. Request was missing a valid JWT as part of request.
Failure	415 Unsupported Media Type	If the request payload provides a media type other than JSON during entity creation request.

### 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.
- The `self` attribute will contain the live link to the REST resource corresponding to this boat. In other words, this is the URL to get this newly created boat. You must not store the `self` attribute in Datastore

#### Success

Status: 201 Created

```
{
  "id": "abc123",
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "loads": [],
  "owner": "unique_jwt_sub_value_eg_1afesaf319fs"
  "self": "https://<your-app>/marina/boats/abc123"
}
```

#### Failure

Status: 400 Bad Request

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

Status: 401 Unauthorized

```
{  
  "Error": "Unauthorized Access error, no valid JWT was provided"  
}
```

Status: 415 Bad Request

```
{  
  "Error": "The request object contains data in an unsupported media type"  
}
```

## Get a Boat

Allows you to get an existing boat. Resources are protected based on owner. Need to provide valid JWT to access.

GET /boats/:boat\_id

### Request

#### Path Parameters

Name	Type	Description
boat_id	String	ID of the boat

#### Request Body

None

### Response

#### Response Header Example

```
{
  "Content-Type": "application/json; charset=utf-8"
}
```

#### Response Body Format

JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	Protected resource requires authorization via valid bearer token in order to access. Request was missing a valid JWT as part of request.
Failure	403 Forbidden	No boat with this boat_id exists, or the resources belongs to another user, and you do not have the authorization to access this resource.
Failure	406 Not Accepted	Request asks for a response in an unsupported content type

#### Response Examples

- Response body contains 'load' attribute which contains JSON array of loads being carried by the boat. Loads will be represented as an id and a 'self' path to the load resource. If no loads currently on boat, then value will be an empty array [ ].

#### Success

Status: 200 OK

```
{
  "id": "abc123",
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
```

```
"loads": [  
  {"id": 5678,  
   "self": "https://<your-app>/marina/loads/5678"},  
  {"id": 8765,  
   "self": "https://<your-app>/marina/loads/8765"}  
],  
"owner": "unique_jwt_sub_value_eg_1afesaf319fs "  
"self": "https://<your-app>/marina/boats/abc123"  
}
```

#### *Failure*

Status: 401 Unauthorized

```
{  
  "Error": "Unauthorized Access error, no valid JWT was provided"  
}
```

Status: 403 Forbidden

```
{  
  "Error": "No entity (e.g. user, boat, or load) with this id exists"  
}
```

Status: 406 Not Accepted

```
{  
  "Error": "API Endpoint does not support the requested response content type"  
}
```



## List all Boats

List all the boats. Utilizes pagination functionality to display a max of 5 boats per page. Return result provides a 'next' URL address to the next set of results. Pagination utilizes cursor methodology to track.

Resources are protected based on owner. Need to provide valid JWT to access.

GET /boats

### Request

Path Parameters

None

Request Body

None

### Response

Response Header Example

```
{
  "Content-Type": "application/json; charset=utf-8"
}
```

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	Protected resource requires authorization via valid bearer token in order to access. Request was missing a valid JWT as part of request.
Failure	406 Not Accepted	Request asks for a response in an unsupported content type

Response Examples

- Response body contains the `loads` attribute which contains JSON array of loads being carried by the boat. Loads will be represented as an `id` and a `self` path to the load resource.
- Pagination: Listing all boats will utilize pagination with a limit of 5 boats per page. Within the response body, there will be a key `next` with a value containing a URL to the next set of results. There will also be a `total_rows` value which informs how many records are related to a specific query.

Success

Status: 200 OK

```
{
  "results": [
```

```

{
  "id": "abc123",
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "loads": [
    {
      "id": 5678,
      "self": "https://<your-app>/loads/5678"
    }
  ],
  "owner": "uniqueidhomerabc123",
  "self": "https://<your-app>/marina/boats/abc123"
},
{
  "id": "def456",
  "name": "Adventure",
  "type": "Sailboat",
  "length": 50,
  "loads": [],
  "owner": "margaretid1234unique",
  "self": "https://<your-app>/marina/boats/abc123"
}
],
"next": "https://<your-app>/marina/boats?cursor=unique_cursor_hash_value",
"total_rows": 10
}

```

#### Failure

Status: 401 Unauthorized

```

{
  "Error": "Unauthorized Access error, no valid JWT was provided"
}

```

Status: 406 Not Accepted

```

{
  "Error": "API Endpoint does not support the requested response content type"
}

```

## Delete a Boat

Allows you to delete a boat. Note that if the boat currently holds a load, those loads are unloaded from the boat and the load carrier values will be updated. Additionally, deleted boats will be removed from the owner's user entity log of boats owned.

Resources are protected based on owner. Need to provide valid JWT to access.

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

### Request

#### Path Parameters

Name	Type	Description
boat_id	String	ID of the boat

### Request Body

None

### Response

#### Response Header Example

```
{
  "Content-Type": "application/json; charset=utf-8"
}
```

### Response Body Format

Success: No body

Failure: JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	Protected resource requires authorization via valid bearer token in order to access. Request was missing a valid JWT as part of request.
Failure	403 Forbidden	No boat with this boat_id exists, or the resources belongs to another user, and you do not have the authorization to access this resource.

### Response Examples

#### Success

```
Status: 204 No Content
```

#### Failure

```
Status: 401 Unauthorized
{
```

<pre>"Error": "Unauthorized Access error, no valid JWT was provided" }</pre>
<pre>Status: 403 Forbidden {   "Error": "No entity (e.g. user, boat, or load) with this id exists" }</pre>

## Edit a Boat with Patch

Allows you to edit any subset of attributes for a boat. Other attributes not part of the request will remain unchanged. Updates to the name attribute will cascade to all related loads and user entities as well.

Resources are protected based on owner. Need to provide valid JWT to access this endpoint for a specific resource.

PATCH /boats/:boat_id
-----------------------

### Request

#### Path Parameters

Name	Type	Description
boat_id	String	ID of the boat

Request Header Mime-Type / Content-Type  
application/json (required)

### Request Body

Required

#### Request Body Format

JSON

#### Request JSON Attributes

Name	Type	Description	Required?
name	String	The name of the boat.  Input Validation / Restrictions: <ul style="list-style-type: none"><li>Name string can only contain alphanumeric chars and spaces. e.g. a-z, A-Z, 0-9 and ' '. Any other input is invalid.</li><li>Name string must be less than 40 characters long. Longer inputs will register as invalid.</li></ul>	No
type	String	The type of the boat. E.g., Sailboat, Catamaran, etc.  Input Validation / Restrictions: <ul style="list-style-type: none"><li>Type string must be less than 100 characters long. Longer inputs will register as invalid.</li></ul>	No
length	Integer	Length of the boat in feet.  Input Validation / Restrictions: <ul style="list-style-type: none"><li>Length value must be non-negative integer. Any trailing decimals will be truncated.</li></ul>	No

### Request Body Example

If additional attributes are included in the request body outside of the standard name, type, and length attributes, these attributes will be dropped from the request body. The marina API will only manage and track data for the name, type, and length attributes.

```
{
  "name": "Sea Witch",
  "length": 99
}
```

## Response

### Response Header Example

```
{
  "Content-Type": "application/json; charset=utf-8"
}
```

## Response Body Format

### JSON

### Response Statuses

Outcome	Status Code	Notes
Success	200 Ok	
Failure	400 Bad Request	Attribute input values will be validated for correctness prior to entity creation. Extraneous attributes will be dropped and ignored during entity creation.
Failure	403 Forbidden	No boat with this boat_id exists, or the resources belongs to another user, and you do not have the authorization to access this resource.
Failure	404 Not Found	No boat with this boat_id exists
Failure	415 Unsupported Media Type	If the request payload provides a media type other than JSON during entity creation request.

## Response Examples

- The `self` attribute will contain the live link to the REST resource corresponding to this boat. You must not store this attribute in Datastore.

### Success

```
Status: 200 OK

{
  "id": "abc123",
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 99,
  "self": "https://<your-app>/boats/abc123"
}
```

### Failure

```
Status: 400 Bad Request

{
  "Error": "The request object is missing at least one of the required attributes or provides an invalid attribute value."
}
```

<pre> } </pre>
<p>Status: 403 Forbidden</p> <pre> {   "Error": " No entities with this entity_id exists or this entity is owned by someone else" } </pre>
<p>Status: 404 Not Found</p> <pre> {   "Error": " No entity (e.g. user, boat, or load) with this id exists" } </pre>
<p>Status: 415 Unsupported Media Type</p> <pre> {   "Error": "The request object contains data in an unsupported media type" } </pre>

## Edit a Boat with Put

Allows you to edit a boat. Updates to the name attribute will cascade to all related loads and user entities as well.

PUT /boats/:boat\_id

### Request

#### Path Parameters

Name	Type	Description
boat_id	String	ID of the boat

#### Request Header Mime-Type / Content-Type

application/json (required)

#### Request Body

Required

#### Request Body Format

JSON

#### Request JSON Attributes

Name	Type	Description	Required?
name	String	The name of the boat.  Input Validation / Restrictions: <ul style="list-style-type: none"><li>Name string can only contain alphanumeric chars and spaces. e.g. a-z, A-Z, 0-9 and ' '. Any other input is invalid.</li><li>Name string must be less than 40 characters long. Longer inputs will register as invalid.</li></ul>	Yes
type	String	The type of the boat. E.g., Sailboat, Catamaran, etc.  Input Validation / Restrictions: <ul style="list-style-type: none"><li>Type string must be less than 100 characters long. Longer inputs will register as invalid.</li></ul>	Yes
length	Integer	Length of the boat in feet.  Input Validation / Restrictions: <ul style="list-style-type: none"><li>Length value must be non-negative integer. Any trailing decimals will be truncated.</li></ul>	Yes

#### Request Body Example

If additional attributes are included in the request body outside of the standard name, type, and length attributes, these attributes will be dropped from the request body. The marina API will only manage and track data for the name, type, and length attributes.

```
{
  "name": "Sea Witch",
  "type": "Catamaran",
```



```
"length": 99
}
```

## Response

### Response Header Example

```
{
  "Content-Type": "application/json; charset=utf-8"
}
```

## Response Body Format

### JSON

### Response Statuses

Outcome	Status Code	Notes
Success	303 See Other	Provides URL of updated boat under the 'Location' header of html response.
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.  Attribute input values will also be validated for correctness prior to entity creation. Extraneous attributes will be dropped and ignored during entity creation.
Failure	401 Unauthorized	Protected resource requires authorization via valid bearer token in order to access. Request was missing a valid JWT as part of request.
Failure	403 Forbidden	No boat with this boat_id exists, or the resources belongs to another user, and you do not have the authorization to access this resource.
Failure	404 Not Found	No boat with this boat_id exists
Failure	415 Unsupported Media Type	If the request payload provides a media type other than JSON during entity creation request.

### Response Examples

- The `self` attribute will contain the live link to the REST resource corresponding to this boat. You must not store this attribute in Datastore.

#### Success

```
Status: 303 See Other
```

#### Failure

```
Status: 400 Bad Request

{
```

<pre>"Error": "The request object is missing at least one of the required attributes or provides an invalid attribute value." }</pre>
<pre>Status: 401 Unauthorized { "Error": "Unauthorized Access error, no valid JWT was provided" }</pre>
<pre>Status: 403 Forbidden { "Error": " No entities with this entity_id exists or this entity is owned by someone else" }</pre>
<pre>Status: 404 Not Found { "Error": " No entity (e.g. user, boat, or load) with this id exists" }</pre>
<pre>Status: 415 Unsupported Media Type { "Error": "The request object contains data in an unsupported media type" }</pre>

## 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	Type	Description	Required?
volume	Integer	Represents the volume of a load.	Yes
content	String	Brief description of the contents of the load  Input Validation / Restrictions: <ul style="list-style-type: none"><li>Name string can only contain alphanumeric chars and spaces. e.g. a-z, A-Z, 0-9 and ' '. Any other input is invalid. Name string must be less than 150 characters long. Longer inputs will register as invalid.</li></ul>	Yes

Request Body Example

```
{
  "volume": 5,
  "content": "lego blocks"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is missing the any attributes, e.g. volume or content, the load will 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.
Failure	415 Unsupported Media Type	If the request payload provides a media type other than JSON during entity creation request.

## 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 which currently holds the load. If the load is currently unloaded at the dock, then the value of `carrier` should be null. On creation, loads are at the dock and not on a boat.
- The value of the attribute `self` is a live link to the REST resource corresponding to this load. In other words, this is the URL to get this newly created load. You must not store this attribute in Datastore.

### Success

Status: 201 Created

```
{
  "id": "123abc",
  "volume": 5,
  "carrier": null,
  "content": "lego blocks",
  "creation_date": "10/18/2020",
  "self": "https://<your-app>/marina/loads/123abc"
}
```

### Failure

Status: 400 Bad Request

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

Status: 415 Unsupported Media Type

```
{
  "Error": "The request object contains data in an unsupported media type"
}
```

## Get a Load

Allows you to get an existing load.

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

### Request

#### Path Parameters

Name	Type	Description
load_id	String	ID of the load

### Response Header Example

<pre>{   "Content-Type": "application/json; charset=utf-8" }</pre>
--

### 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
Failure	406 Not Accepted	Request asks for a response in an unsupported content type

### Response Examples

#### Success

<p>Status: 200 OK</p> <pre>{   "id": "123abc",   "volume": 5,   "content": "lego blocks",   "carrier": {     "id": 1234,     "name": "Sea Witch",     "self": "https://&lt;your-app&gt;/boats/1234"   },   "creation_date": "01/01/2021",   "self": "https://&lt;your-app&gt;/loads/123abc" }</pre>
---

#### Failure

Status: 404 Not Found
-----------------------

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

Status: 406 Not Found

```
{  
  "Error": "API Endpoint does not support the requested response content type"  
}
```

## List all Loads

List all the loads. Utilizes pagination functionality to display a max of 5 loads per page. Return result provides a 'next' URL address to the next set of results. Pagination utilizes cursor methodology to track.

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

### Request

Path Parameters

None

Request Body

None

### Response

Response Body Format

JSON

### Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Accepted	Request asks for a response in an unsupported content type

### Response Examples

- Response body contains the `carrier` attribute which contains JSON object of the carrier boat. `Carrier` will be represented as an `id`, `name`, and a `self` path to the carrier resource.
- Pagination: Listing all boats will utilize pagination with a limit of 5 boats per page. Within the response body, there will be a key `next` with a value containing a URL to the next set of results. There will also be a `total_rows` value which informs how many records are related to a specific query.

### Success

Status: 200 OK

```
{
  "results": [
    {
      "id": "abc123",
      "volume": 5
      "content": "lego blocks",
      "carrier": {
        "id": 1234,
        "name": "Sea Witch",
        "self": "https://<your-app>/marina/boats/1234"
      },
      "creation_date": "01/01/2021",
      "self": "https://<your-app>/marina/loads/abc123"
    },
  ],
}
```

```
{
  "id": "123abc",
  "volume": 10
  "content": "jena blocks",
  "carrier": null,
  "creation_date": "01/01/2021",
  "self": "https://<your-app>/marina/loads/123abc"
},
],
"next": "https://<your-app>/marina/loads?cursor=unique_cursor_hash_value",
"total_rows": 10
}
```

#### *Failure*

Status: 406 Not Accepted

```
{
  "Error": "API Endpoint does not support the requested response content type"
}
```



## Delete a Load

Allows you to delete a load. If the load being deleted is currently on a boat, the boat will be updated to remove the load as well.

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

### Request

#### Path Parameters

Name	Type	Description
load_id	String	ID of the load

### Request Body

None

### Response

#### Response Body Format

Success: No body

Failure: JSON

#### Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	403 Not Found	No load with this load_id exists

### Response Examples

#### Success

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

#### Failure

Status: 403 Forbidden { "Error": No entity (e.g. user, boat, or load) with this id exists }
--

## Edit a Load with Patch

Allows you to edit any subset of attributes for a load. Other attributes not part of the request will remain unchanged.

PATCH /loads/:load_id
-----------------------

### Request

#### Path Parameters

Name	Type	Description
load_id	String	ID of the load

#### Request Header Mime-Type / Content-Type

application/json (required)

### Request Body

Required

#### Request Body Format

JSON

#### Request JSON Attributes

Name	Type	Description	Required?
context	String	Details on what is contained in the load.  Input Validation / Restrictions: <ul style="list-style-type: none"><li>Name string can only contain alphanumeric chars and spaces. e.g. a-z, A-Z, 0-9 and ' '. Any other input is invalid.</li><li>Name string must be less than 150 characters long. Longer inputs will register as invalid.</li></ul>	No
volume	Integer	Length of the boat in feet.  Input Validation / Restrictions: <ul style="list-style-type: none"><li>Length value must be non-negative integer. Any trailing decimals will be truncated.</li></ul>	No

### Request Body Example

If additional attributes are included in the request body outside of the standard name, type, and length attributes, these attributes will be dropped from the request body. The marina API will only manage and track data for the name, type, and length attributes.

<pre>{   "volume": 126,   "content": "Pokemon Cards" }</pre>
--

## Response

### Response Header Example

```
{
  "Content-Type": "application/json; charset=utf-8"
}
```

## Response Body Format

### JSON

### Response Statuses

Outcome	Status Code	Notes
Success	200 Ok	
Failure	400 Bad Request	Attribute input values will be validated for correctness prior to entity creation. Extraneous attributes will be dropped and ignored during entity creation.
Failure	404 Not Found	No load with this load_id exists
Failure	415 Unsupported Media Type	If the request payload provides a media type other than JSON during entity creation request.

### Response Examples

- The `self` attribute will contain the live link to the REST resource corresponding to this boat. You must not store this attribute in Datastore.

#### Success

```
Status: 200 OK

{
  "id": "abc123",
  "volume": 127,
  "content": "Light Sabers",
  "carrier": {
    "id": 1234,
    "name": "Sea Witch",
    "self": "https://<your-app>/marina/boats/1234"
  },
  "creation_date": "01/01/2021",
  "self": "https://<your-app>/marina/loads/abc123"
}
```

#### Failure

```
Status: 400 Bad Request

{
  "Error": "The request object is missing at least one of the required attributes or provides an invalid attribute value."
}

Status: 404 Not Found
```

```
{  
  "Error": "No entity (e.g. user, boat, or load) with this id exists"  
}
```

Status: 415 Unsupported Media Type

```
{  
  "Error": "The request object contains data in an unsupported media type"  
}
```

## Edit a Load with Put

Allows you to edit a Load.

```
PUT /loads/:load_id
```

### Request

#### Path Parameters

Name	Type	Description
load_id	String	ID of the load

#### Request Header Mime-Type / Content-Type

application/json (required)

### Request Body

Required

#### Request Body Format

JSON

#### Request JSON Attributes

Name	Type	Description	Required?
context	String	Details on what is contained in the load.  Input Validation / Restrictions: <ul style="list-style-type: none"><li>Name string can only contain alphanumeric chars and spaces. e.g. a-z, A-Z, 0-9 and ' '. Any other input is invalid.</li><li>Name string must be less than 150 characters long. Longer inputs will register as invalid.</li></ul>	Yes
volume	Integer	Length of the boat in feet.  Input Validation / Restrictions: <ul style="list-style-type: none"><li>Length value must be non-negative integer. Any trailing decimals will be truncated.</li></ul>	Yes

### Request Body Example

If additional attributes are included in the request body outside of the standard name, type, and length attributes, these attributes will be dropped from the request body. The marina API will only manage and track data for the name, type, and length attributes.

```
{
  "volume": 126,
  "content": "Pokemon Cards"
}
```

### Response

#### Response Header Example

```
{
  "Content-Type": "application/json; charset=utf-8"
}
```

```
}
```

## Response Body Format

### JSON

## Response Statuses

Outcome	Status Code	Notes
Success	303 See Other	Provides URL of updated boat under the 'Location' header of html response.
Failure	400 Bad Request	Attribute input values will be validated for correctness prior to entity creation. Extraneous attributes will be dropped and ignored during entity creation.
Failure	404 Not Found	No load with this load_id exists
Failure	415 Unsupported Media Type	If the request payload provides a media type other than JSON during entity creation request.

## Response Examples

- The `self` attribute will contain the live link to the REST resource corresponding to this boat. You must not store this attribute in Datastore.

### Success

Status: 303 See Other

### Failure

Status: 400 Bad Request
<pre>{   "Error": "The request object is missing at least one of the required attributes or   provides an invalid attribute value." }</pre>
Status: 404 Not Found
<pre>{   "Error": " No entity (e.g. user, boat, or load) with this id exists" }</pre>
Status: 415 Unsupported Media Type
<pre>{   "Error": "The request object contains data in an unsupported media type" }</pre>

## Assign Load to a Boat

Adding a load onto a boat. The load has been assigned to a carrier. The boat's log of onboard loads is updated as well as the load's carrier value.

PUT /boats/:boat\_id/loads/:load\_id

### Request

#### Path Parameters

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

### Request Body

None

### Response

#### 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 currently is not on another boat.
Failure	403 Forbidden	There is already a carrier boat for that load.
Failure	404 Not Found	No boat with this boat_id exists, and/or no load with this load_id exists.

### Response Examples

#### Success

Status: 204 No Content

#### Failure

Status: 403 Forbidden
<pre>{   "Error": "The load is already on another boat " }</pre>
Status: 404 Not Found
<pre>{   "Error": "The specified boat and/or load does not exist" }</pre>

### Comment

- A load cannot be assigned to multiple boats.

## Remove Load from a Boat

Removes a load from a boat. The load can be reassigned to another carrier and has a carrier value of null. The boat's log of onboard loads is updated.

DELETE /boats/:boat_id/loads/:load_id
---------------------------------------

### Request

#### Path Parameters

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

#### Request Body

None

### Response

#### 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 on this boat.
Failure	404 Not Found	No load with this load_id is on the boat with this boat_id.

#### Response Examples

##### Success

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

##### Failure

Status: 404 Not Found
<pre>{   "Error": "No load with this load_id is on the boat with this boat_id" }</pre>



## List all Users

List all the Users. Utilizes pagination functionality to display a max of 5 users per page. Return result provides a 'next' URL address to the next set of results. Pagination utilizes cursor methodology to track.

Resource is unprotected and available for public consumption.

GET /users

### Request

Path Parameters

None

Request Body

None

### Response

Response Header Example

```
{
  "Content-Type": "application/json; charset=utf-8"
}
```

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Accepted	Request asks for a response in an unsupported content type

### Response Examples

- Response body contains the `boats` attribute which contains JSON array of boats that are owned by a user. Boats will be represented as an `id`, `name`, and a `self` path to the boat resource.
- Pagination: Listing all boats will utilize pagination with a limit of 5 boats per page. Within the response body, there will be a key `next` with a value containing a URL to the next set of results. There will also be a `total_rows` value which informs how many records are related to a specific query.

#### Success

```
Status: 200 OK

{
  "results": [
    {
      "id": "5184813602439168",
      "nickname": "luffy",
```

```

    "name": "Monkey D. Luffy",
    "email": "luffy@onepiece.com",
    "auth0_id": "auth0|609df7965f091d0069943e9a",
    "boats": [
      {
        "id": "6596285448585216",
        "name": "Swordfish",
        "self": "https://<your-app>/marina/boats/6596285448585216"
      }
    ],
    "self": "https://<your-app>/marina/users/5184813602439168"
  },
  {
    "id": "5747763555860480",
    "nickname": "spike",
    "name": "Spike Spiegel",
    "email": "spike@bebop.com",
    "auth0_id": "auth0|609df721cbc3e700700dc1c5",
    "boats": [],
    "self": "https://<your-app>/marina/users/5747763555860480"
  }
],
"total_rows": 2
}

```

#### Failure

Status: 406 Not Accepted

```

{
  "Error": "API Endpoint does not support the requested response content type"
}

```

