

# Portfolio Assignment: API Spec

CS 493: Cloud Application Development

Fall 2022

Oregon State University

Last Update: Dec 4 11:59 pm Pacific

App Url: <https://cs493-portfolio-saechaok.wl.r.appspot.com>

Login Url: <https://cs493-portfolio-saechaok.wl.r.appspot.com>

Change log .....	1
Data Model .....	2
Relationships.....	2
Boat Attribute Constraints.....	2
Load Attribute Constraints.....	4
User Entity Model Description.....	5
User Attribute Constraints.....	5
Create a Boat .....	6
View all Boats — Protected .....	8
View all Boats — Unprotected.....	10
View a Boat.....	12
Edit a Boat — Patch Request .....	14
Edit a Boat — Put Request.....	17
Delete a Boat .....	20
Assign Load to a Boat – Put Request .....	22
Delete Load From a Boat .....	23
Create a Load .....	24
View a Load.....	26
View all Loads .....	27
Delete a Load .....	29
View all Users.....	30

## Change log

Version	Change	Date
---------	--------	------

1.0	Initial version.	Dec 4, 2022
-----	------------------	-------------

## Data Model

The app stores **three** different entities in Datastore: Boats, Loads, and Users.

## Relationships

### Boats ↔ Loads

The boats and loads entities are related such that loads can be on the boats. If a load is on a boat then, that particular boat will contain an object representation of that load. The object will contain the load id, and the 'self' link directing you to that load resource.

On the other side, loads will contain the property carrier, with the value of that property being an object containing the boat id and the 'self' link directing you to the boat carrying that load.

### Boats ↔ Users

Boats and Users entities are related such that users can own boats. If a boat is owned by a user, then the boat's owner property will contain the value of an object containing the user id and the 'self' link directing you to the user that owns the boat.

On the other side, if a user owns a particular boat, then the user's owned\_boats property will contain an object in the array. The object will contain the boat id and the 'self' link directing you to the boat owned.

### Boats

Property	Data Type	Notes	Required
id	Integer	The id of the boat. Datastore automatically generates it.	No*
name	String	Name of the boat.	Yes
type	String	Type of the boat. E.g. Sailboat, Catamaran, etc.	Yes
length	Integer	The length of the boat in feet.	Yes
loads	Array	List of loads on the boat	No*
owner	String	Owner of the boat represented by an owner id	No*

\*Note: attributes where required = No, means it will be generated for you, and not required in the request body.

## Boat Attribute Constraints

### Name Constraints

The name attribute must be valid. Invalid names will result in a 400 status error. Follow the restrictions below:

- Name must be a string
- Name cannot be an empty string
- Names cannot be longer than 20 characters

### Type Constraints

The boat type attribute must be valid. Invalid types will result in a 400 status error. Follow the restrictions below:

- Type must be a string
- Type cannot be an empty string
- Type cannot be longer than 20 characters

### Length Constraints

The boat length attribute must be valid. Invalid length will result in a 400 status error. Follow the restrictions below:

- Length must be greater than 0.

## Loads

Property	Data Type	Notes	Required
id	Integer	The id of the load. Datastore automatically generates it.	No*
item	String	Name of the load.	Yes
creation_date	String	Date when the load was created	Yes
volume	Integer	The volume of the load in cubic feet.	Yes
carrier	Object	The boat that holds this load	No*

\*Note: attributes where required = No, means it will be generated for you, and not required in the request body.

## Load Attribute Constraints

### item Constraints

The name attribute must be valid. Invalid names will result in a 400 status error. Follow the restrictions below:

- Item must be a string
- Item cannot begin with a space
- Item cannot be an empty string
- item cannot be longer than 20 characters

### Volume Constraints

The boat type attribute must be valid. Invalid types will result in a 400 status error. Follow the restrictions below:

- volume must be an integers
- volume must be greater than 0

### Length Constraints

- creation\_date must be a string

## Users

Property	Data Type	Notes	Required
id	Integer	The id of the load. Datastore automatically generates it.	No*
name	String	Name of the user, extracted from decoded JWT	No*
userId	String	User id of the user, extracted from decoded JWT	No*
Owned_boats	Array	Array will hold the boats owned by the user	No*

\*Note: attributes where required = No, means it will be generated for you, and not required in the request body.

## User Entity Model Description

What is the unique identifier for a user in your data model?

- The unique identifier for each user is the 'sub' property extracted from the decoded JWT supplied by Auth0. This will be known as the userId.

If a request needs to supply a user identifier, what needs to be specified by the person making the request.

- All protected endpoints will require the 'Authorization Header'. You will need to supply the jwt id\_token as a 'Bearer token'.

How your application maps a supplied JWT to the identifier of a user.

- The supplied jwt passed into the protected endpoint will be decoded and its 'sub' value extracted. This 'sub' value needs to be mapped to the 'userId' identifier saved in the Datastore.

What is the relationship between the user entity and non-user entity.

- The user entity will share a relationship with the Boats entity. Users will own boats, and boats will be owned by users.

## User Attribute Constraints

\* None.

\*Note: user attributes will be automatically generated by decoding the Bearer token provided in the form of a JWT and the datastore.

## Create a Boat

Allows you to create a new boat.

POST /boats

Request

Path Parameters

None

Authorization

Bearer Token

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*

\* Please see attribute constraints on page 2

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	There is an error in the request
Failure	401 Unauthorized	Missing or invalid credentials
Failure	406 Not Acceptable	The response type requested is not supported
Failure	415 Unsupported Media Type	The request type is not supported

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.
- Response body will contain "self" links. The links will go to the most direct location to find that resource

- Created boats will have an owner property with the value corresponding to the decoded JWT user id, and no loads assigned

#### Success

Status: 201 Created

```
{
  "id": 6000221733519360,
  "name": "Hello World",
  "type": "longboat",
  "length": 10,
  "owner": "63883d1232de0084d4ee2cfe",
  "loads": [],
  "self": "https://cs493-portfolio-saechaok.wl.r.appspot.com/boats/6000221733519360"
}
```

#### Failures

Status: 400 Bad Request

```
{
  "Error": "400 Bad Request",
  "Message": "There is an error in the request"
}
```

Status: 401 Unauthorized

```
{
  "Error": "401 Unauthorized",
  "Message": "Missing or invalid credentials"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "406 Not Acceptable",
  "Message": "Server cannot provide requested media type"
}
```

Status: 415 Unsupported Media Type

```
{
  "Error": "415 Unsupported Media Type",
  "Message": "Server cannot accept media type"
}
```

## View all Boats — Protected

List all the boats.

GET /boats

Request

Path Parameters

None

Authorization

Bearer Token

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

Response Examples

- The ***protected*** get all route will return all boats that belong to the user id in the decoded JWT
- A successful response will contain the result of the query and the total boats currently stored in the "total\_collection\_length" property that belong to the user id in the JWT.
- Response body will contain "self" links. The links will go to the most direct location to find that resource
- Response body will paginate the GET request to return 5 boats at a time per page. The "next" property will paginate the next 5 boats until no more boats are remaining in the datastore. At that point there will be no "next" property.
- The 'next' and 'prev' link will contain a 'cursor' that will direct you to the 'prev' 5 boats or to the 'next' 5 boats until you reach the end of the datastore. The cursors remembers what thing it last returned and what the next thing is to grab, so it is unaffected by inserts or deletes of things it already retrieved.

Success

Status: 200 OK

```
{
  "total_collection_length": 9,
  "result": [
    {
      "loads": [],
      "type": "shortboat",
```



```

    "id": 4829542061441024,
    "owner": "63883d1232de0084d4ee2cfe",
    "length": 2,
    "name": "Hello World 2",
    "self": "https://cs493-portfolio-saechaok.wl.r.appspot.com/boats/4829542061441024"
  },
  {
    "length": 10,
    "type": "longboat",
    "owner": "63883d1232de0084d4ee2cfe",
    "loads": [],
    "name": "Hello World",
    "id": 5165683113459712,
    "self": "https://cs493-portfolio-saechaok.wl.r.appspot.com/boats/5165683113459712"
  },
  {
    "type": "longboat",
    "owner": "63883d1232de0084d4ee2cfe",
    "loads": [],
    "name": "Hello World",
    "length": 10,
    "id": 5171534335311872,
    "self": "https://cs493-portfolio-saechaok.wl.r.appspot.com/boats/5171534335311872"
  },
  {
    "length": 10,
    "name": "Hello World",
    "type": "longboat",
    "loads": [],
    "owner": "63883d1232de0084d4ee2cfe",
    "id": 5199107488481280,
    "self": "https://cs493-portfolio-saechaok.wl.r.appspot.com/boats/5199107488481280"
  },
  {
    "name": "Hello World",
    "type": "longboat",
    "length": 10,
    "loads": [],
    "owner": "63883d1232de0084d4ee2cfe",
    "id": 5722078678351872,
    "self": "https://cs493-portfolio-saechaok.wl.r.appspot.com/boats/5722078678351872"
  }
],
"next": "https://cs493-portfolio-saechaok.wl.r.appspot.com/boats?cursor=CjYSMGoabX5jczQ5My1wb3J0Zm9saW8tc2FIY2hnb2tyEgsBUjVYXRzGICAgJi0hpUKDBgAIAA="
}

```

## View all Boats — Unprotected

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

- The **protected** get all route will return all boats that have been created
- A successful response will contain the result of the query and the total boats currently stored in the "total\_collection\_length" property.
- Response body will contain "self" links. The links will go to the most direct location to find that resource
- Response body will paginate the GET request to return 5 boats at a time per page. The "next" property will paginate the next 5 boats until no more boats are remaining in the datastore. At that point there will be no "next" property.
- The "next" and "prev" link will contain a "cursor" that will direct you to the "prev" 5 boats or to the "next" 5 boats until you reach the end of the datastore. The cursors remembers what thing it last returned and what the next thing is to grab, so it is unaffected by inserts or deletes of things it already retrieved.

Success

Status: 200 OK

```
{
  "total_collection_length": 9,
  "result": [
    {
      "type": "shortboat",
      "loads": [],
      "length": 2,
      "owner": "12343d1232de0084d4ee2cfe",
      "id": 4829542061441024,
```

```

    "name": "Hello World 2",
    "self": "https://cs493-portfolio-saechaok.wl.r.appspot.com/boats/4829542061441024"
  },
  {
    "length": 10,
    "type": "longboat",
    "loads": [],
    "name": "Hello World",
    "owner": "54321d1232de0084d4ee2cfe",
    "id": 5165683113459712,
    "self": "https://cs493-portfolio-saechaok.wl.r.appspot.com/boats/5165683113459712"
  },
  {
    "length": 10,
    "owner": "78973d1232de0084d4ee2cfe",
    "name": "Hello World",
    "type": "longboat",
    "loads": [],
    "id": 5171534335311872,
    "self": "https://cs493-portfolio-saechaok.wl.r.appspot.com/boats/5171534335311872"
  },
  {
    "loads": [],
    "type": "longboat",
    "name": "Hello World",
    "owner": "43613d1232de0084d4ee2cfe",
    "length": 10,
    "id": 5199107488481280,
    "self": "https://cs493-portfolio-saechaok.wl.r.appspot.com/boats/5199107488481280"
  },
  {
    "type": "longboat",
    "name": "Hello World",
    "loads": [],
    "owner": "63883d1232de0084d4ee2cfe",
    "length": 10,
    "id": 5722078678351872,
    "self": "https://cs493-portfolio-saechaok.wl.r.appspot.com/boats/5722078678351872"
  }
],
"next": "https://cs493-portfolio-saechaok.wl.r.appspot.com/boats?cursor=CjYSMGoabX5jczQ5My1wb3J0Zm9saW8tc2FIY2hnb2tyEgsBUjVYXRzGICAgJi0hpUKDBgAIAA="
}

```

## 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

Bearer Token

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	Missing or invalid credentials
Failure	403 Forbidden	Invalid credentials for the resource
Failure	404 Not Found	no resource with this id exists
Failure	406 Not Acceptable	Invalid requested media type

Response Examples

- Response body will contain "self" links. The links will go to the most direct location to find that resource

*Success*

Status: 200 OK

```
{
  "length": 10,
  "loads": [],
  "owner": "63883d1232de0084d4ee2cfe",
  "type": "longboat",
  "name": "Hello World",
  "id": 6000221733519360,
  "self": "https://cs493-portfolio-saechaok.wl.r.appspot.com/boats/6000221733519360"
}
```

### Failure

<p>Status: 401 Unauthorized</p> <pre>{   "Error": "401 Unauthorized",   "Message": "Missing or invalid credentials" }</pre>
<p>Status: 403 Unauthorized</p> <pre>{   "Error": "403 Forbidden",   "Message": "Invalid credentials for the resource" }</pre>
<p>Status: 404 Not Found</p> <pre>{   "Error": "404 Not Found",   "Message": "no resource with this id exists" }</pre>
<p>Status: 406 Not Acceptable</p> <pre>{   "Error": "406 Not Acceptable",   "Message": "Server cannot provide media type" }</pre>

## Edit a Boat — Patch Request

Edit a boat

PATCH /boats/:boat\_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Authorization

Bearer Token

Request Body

Required

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*

\* Please see attribute constraints on page 2

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	The contents of request body is incorrect.
Failure	401 Unauthorized	Missing or invalid credentials
Failure	403 Forbidden	Invalid credentials for the resource
Failure	404 Not Found	No resource with this id exists
Failure	406 Not Acceptable	The response type requested is not supported
Failure	415 Unsupported Media Type	The request type is not supported

Response Examples

*Success*

Status: 200 OK
<pre>{   "owner": "63883d1232de0084d4ee2cfe",   "loads": [],   "name": "Hello World 2",</pre>

```
"type": "longboat",  
"length": 10,  
"id": 4829542061441024,  
"self": "https://cs493-portfolio-saechaok.wl.r.appspot.com/boats/4829542061441024"  
}
```

#### *Failure*

Status: 400 bad Request

```
{  
  "Error": "400 Bad Request",  
  "Message": "There is an error in the request"  
}
```

Status: 401 Unauthorized

```
{  
  "Error": "401 Unauthorized",  
  "Message": "Missing or invalid credentials"  
}
```

Status: 403 Forbidden

```
{  
  "Error": "403 Forbidden",  
  "Message": "Invalid credentials for the resource"  
}
```

Status: 404 Not Found

```
{  
  "Error": "404 Not Found",  
  "Message": "No resource with this id exists"  
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "406 Not Acceptable",  
  "Message": "Server cannot provide requested media type"  
}
```

Status: 415 Unsupported Media Type

```
{  
  "Error": "415 Unsupported Media Type",  
}
```

```
"Message": "Server cannot accept media type"  
}
```



## Edit a Boat — Put Request

List all the boats.

PUT /boats/:boat\_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Authorization

Bearer Token

Request Body

Required

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*

\* Please see attribute constraints on page 2

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	The contents of request body is incorrect.
Failure	401 Unauthorized	Missing or invalid credentials
Failure	403 Forbidden	Invalid credentials for the resource"
Failure	404 Not Found	No resource with this id exists
Failure	406 Not Acceptable	Server cannot provide requested media type
Failure	415 Unsupported Media Type	Server cannot accept media type

Response Examples

Success

Status: 200 OK

```
{  
  "owner": "63883d1232de0084d4ee2cfe",
```

```
{
  "type": "shortboat",
  "name": "Hello World 3",
  "length": 2,
  "loads": [],
  "id": 4829542061441024,
  "self": "https://cs493-portfolio-saechaok.wl.r.appspot.com/boats/4829542061441024"
}
```

#### Failure

Status: 400 bad Request

```
{
  "Error": "400 Bad Request",
  "Message": "There is an error in the request"
}
```

Status: 401 Unauthorized

```
{
  "Error": "401 Unauthorized",
  "Message": "Missing or invalid credentials"
}
```

Status: 403 Bad Request

```
{
  "Error": "403 Forbidden",
  "Message": "Invalid credentials for the resource"
}
```

Status: 404 Not Found

```
{
  "Error": "404 Not Found",
  "Message": "No resource with this id exists"
}
```

Status: 406 bad Request

```
{
  "Error": "406 Not Acceptable",
  "Message": "Server cannot provide requested media type"
}
```

Status: 415 bad Request

```
{
```

```
"Error": "415 Unsupported Media Type",  
"Message": "Server cannot accept media type"  
}
```

## Delete a Boat

Allows you to delete a boat.

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

### Request

#### Path Parameters

Name	Description
boat_id	ID of the boat

#### Authorization

Bearer Token

### 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	401 Unauthorized	Missing or invalid credentials
Failure	403 Forbidden	Invalid credentials for the resource
Failure	404 Not Found	no resource with this id exists

#### Response Examples

##### Success

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

### *Failure*

Status: 401 Unauthorized

```
{  
  "Error": "401 Unauthorized",  
  "Message": "Missing or invalid credentials"  
}
```

Status 403 Forbidden

```
{  
  "Error": "403 Forbidden",  
  "Message": "Invalid credentials for the resource"  
}
```

Status: 404 Not Found

```
{  
  "Error": "404 Not Found",  
  "Message": "No resource with this id exists"  
}
```

## Assign Load to a Boat – Put Request

Assign a load to a boat

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

Request

Path Parameters

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

Request Body

Required

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*

\* Please see attribute constraints on page 2

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	404 Not Found	Invalid boat or load id

Response Examples

*Success*

Status: 204 No Content

*Failure*

Status: 404 Not Found

```
{
  "Error": "404 Not Found",
  "Message": "No resource with this id exists"
}
```

## Delete Load From a Boat

Allows you to remove a load from a boat

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

### Request

#### Path Parameters

Name	Description
boat_id	ID of the boat
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 resource with this id exists

#### Response Examples

##### Success

Status: 204 No Content

##### Failure

Status: 404 Not Found

```
{
  "Error": "404 Not Found",
  "Message": "No resource with this 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 in cubic feet	Yes
item	The description of the load	Yes
creation_date	The date the load was create	Yes

Request Body Example

```
{
  "volume": 5,
  "item": "LEGO Blocks",
  "creation_date": "10/18/22"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	There is an error in the request
Failure	406 Not Acceptable	Server cannot provide requested media type
Failure	415 Unsupported Media Type	Server cannot accept media type



## Response Examples

- Datastore will automatically generate an ID and store it with the entity being created.
- The value of the attribute `carrier` is the ID of the boat that currently holds this load. If there is no boat assigned to this load, the value of `carrier` should be null.
- Response body will contain "self" links. The links will go to the most direct location to find that resource

### Success

Status: 201 Created

```
{
  "id": 6291583020302336,
  "item": "Legos",
  "volume": 10,
  "creation_date": "1/1/22",
  "carrier": null,
  "self": "https://cs493-portfolio-saechaok.wl.r.appspot.com/loads/6291583020302336"
}
```

### Failure

Status: 400 Bad Request

```
{
  "Error": "400 Bad Request",
  "Message": "There is an error in the request"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "406 Not Acceptable",
  "Message": "Server cannot provide requested media type"
}
```

Status: 415 Unsupported Media Type

```
{
  "Error": "415 Unsupported Media Type",
  "Message": "Server cannot accept media type"
}
```

## View a Load

Allows you to get an existing slip.

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 resource with this id exists

### Response Examples

- Response body will contain "self" links. The links will go to the most direct location to find that resource

#### Success

Status: 200 OK

```
{
  "volume": 10,
  "creation_date": "1/1/22",
  "carrier": null,
  "item": "Legos",
  "id": 6302012408856576,
  "self": "https://cs493-portfolio-saechaok.wl.r.appspot.com/loads/6302012408856576"
}
```

#### Failure

Status: 404 Not Found

```
{
  "Error": "404 Not Found",
  "Message": "no resource with this 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

- Response body will contain "self" links. The links will go to the most direct location to find that resource
- Response body will paginate the GET request to return 5 loads at a time per page. The "next" property will paginate the next 5 loads until no more loads are remaining in the datastore. At that point there will be no "next" property.
- The 'next' and 'prev' link will contain a 'cursor' that will direct you to the 'prev' 5 loads or to the 'next' 5 loads until you reach the end of the datastore. The cursors remembers what thing it last returned and what the next thing is to grab, so it is unaffected by inserts or deletes of things it already retrieved.

Success

Status: 200 OK

```
{
  "total_collection_length": 45,
  "result": [
    {
      "carrier": null,
      "volume": 20,
      "item": "Balls",
      "creation_date": "2/1/22",
      "id": 5088723339313152,
      "self": "https://cs493-portfolio-saechaok.wl.r.appspot.com/loads/5088723339313152"
    },
  ],
}
```

```

    "item": "Legos",
    "carrier": null,
    "id": 5107442887163904,
    "volume": 100,
    "creation_date": "1/1/22",
    "self": "https://cs493-portfolio-saechaok.wl.r.appspot.com/loads/5107442887163904"
  },
  {
    "creation_date": "2/1/22",
    "carrier": null,
    "item": "Balls",
    "volume": 20,
    "id": 5116720486285312,
    "self": "https://cs493-portfolio-saechaok.wl.r.appspot.com/loads/5116720486285312"
  },
  {
    "item": "Balls",
    "creation_date": "2/1/22",
    "carrier": null,
    "volume": 20,
    "id": 5118606849343488,
    "self": "https://cs493-portfolio-saechaok.wl.r.appspot.com/loads/5118606849343488"
  },
  {
    "item": "Legos",
    "carrier": {
      "id": 5631821081477120,
      "name": "Hello World",
      "self": "https://cs493-portfolio-saechaok.wl.r.appspot.com/boats/5631821081477120"
    },
    "volume": 10,
    "creation_date": "1/1/22",
    "id": 5120422009896960,
    "self": "https://cs493-portfolio-saechaok.wl.r.appspot.com/loads/5120422009896960"
  }
],
"next": "https://cs493-portfolio-saechaok.wl.r.appspot.com/loads?cursor=CjYSMGoabX5jcQ5My1wb3J0Zm9saW8tc2FIY2hhb2tyEgsSBUxvYWRzGICAgLjyn4wJDBgAIAA="
}

```

## Delete a Load

Allows you to delete a 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

```
{
  "Error": "404 Not Found",
  "Message": "no resource with this id exists"
}
```

## View all Users

List all the users.

GET /users

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

Response Examples

- Response body will contain "self" links. The links will go to the most direct location to find that resource

Success

```
Status: 200 OK
{
  "result": [
    {
      "userId": "638c16631f2a815a71110853",
      "name": "user3@cheese.com",
      "owned_boats": [],
      "id": 4749321870245888,
      "self": "https://cs493-portfolio-saechaok.wl.r.appspot.com/users/4749321870245888"
    },
    {
      "id": 6245824170295296,
      "userId": "63883d1232de0084d4ee2cfe",
      "name": "wallace@cheese.com",
      "owned_boats": [
        {
          "self": "http://localhost:8080/boats/5661864209940480",
          "id": 5661864209940480
        }
      ],
      "self": "https://cs493-portfolio-saechaok.wl.r.appspot.com/users/6245824170295296"
    }
  ]
}
```

}