

Portfolio Assignment: Final Project

CS 493: Cloud Application Development

Fall 2023

Oregon State University

Matthew Walker

Last Update: Dec 10. 11:15 am Pacific

Deployed URL.....	1
Data Model	2
Relationship between non-user entities.....	3
How user entity is being modeled	3
Create a Boat	4
List all Boats with Pagination	7
Get a Boat	10
Delete a Boat.....	13
Change a Boat (PUT)	15
Update a Boat (PATCH)	18
Load Assigned to Boat.....	21
Unassign/Remove a Load from a Boat.....	23
Create a Load	24
List all Loads with Pagination	26
Get a Load	28
Change a Load (PUT)	30
Update a Load (PATCH).....	32
Delete a Load	34
Get Users.....	35

Deployed URL

Location	URL	Date
GCP	https://final-project-walkerm7.uw.r.appspot.com	Dec 10, 2023
Account creation/Login	https://final-project-walkerm7.uw.r.appspot.com	Dec 10, 2023

Data Model

The app stores three kinds of entities in Datastore: Boats, Loads, and Users.

Boats

Property	Data Type	Required?	Notes
id	Integer	Yes	The id of the boat. Datastore automatically generates it. Don't add it yourself as a property of the entity.
name	String	Yes	Name of the boat.
type	String	Yes	Type of the boat. E.g., Sailboat, Catamaran, etc.
length	Integer	Yes	The length of the boat in feet.
loads	Array of integer Objects	Yes	An array that contains loads associated (loaded) on the given boat. Specifically, an array of objects that contains IDs for loads. (Automatically generated empty no need to create it) (It is appropriate for it to be empty)
Owner	String	Yes	The id (sub) value is generated from JWT and is used for authentication of user resources.

Loads

Property	Data Type	Required?	Notes
id	Integer	Yes	The id of the load. Datastore automatically generates it. Don't add it yourself as a property of the entity.
volume	Integer	Yes	The volume of the load.
item	String	Yes	The item(s) that the load is storing.
creation_date	String	Yes	A string that should be presented in date format representing the date the load was created.
color	String	Yes	A string that represents the color of the load (container).
carrier	Integer Obj	Yes	The id of the boat that the load is stored on as an integer object. Will start out unloaded which is represented as null.

Users

Property	Data Type	Required?	Notes
id	String	Yes	The ID of the user once an account is created. Uses the "sub" value of JWT.
email	String	No	The email that the user used to create an account. Taken automatically on first login.

Comment

- It is important to note that the API generates "self" links of type strings that are the most direct URLs to the specific resource on the fly when getting any of the resources from the data model. These are never stored in the database (data store) and are always manually generated by the API whenever a resource is requested.

- For my implementation of this API users can't be updated/deleted. They are created on first login/signup and their ID is generated from the "sub" value in JWT and their email is taken so it is easier to see what ID belongs to what user.
- It is important to note that the particular use case/situation the API was designed was that no proper data validation is supplied and it is assumed that request body's will have valid input or errors that have been designed to be caught.

Relationship between non-user entities

The non-user entities Boats and Loads model a one-to-many relationship. A single boat can have many loads associated with it that it is carrying. A load can only really be on one ship at a time although this is not tested or checked. To associate these two entities each stores the ID of the associated object. For instance, the boat's property loads contains an array of load IDs, whereas a Load has the property carrier and stores a single Boat ID that the load is associated with or being "carried" by.

How user entity is being modeled

My application uses the "sub" property from the JWT as a unique identifier for the user entity and for keeping track of what entities a user owns or has access to. All the requests that need a user identifier are handled simply by providing a proper JWT. My API uses the JWT directly and extracts the same "sub" value that is stored in the user entity as the ID to handle authentication and provide access to the corresponding entities (in this case boats). My API did not use any routes to directly link the User entity and its associated non-user entity (boats) rather that is taken care of directly via JWT and manipulation of the API takes care of.

Create a Boat

Allows you to create a new boat. A boat is a protected endpoint so a valid JWT must be provided.

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	<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>For this use case no validation of provided values for the attributes of the request is performed. It is assumed that if the request contains any of the listed attributes, then that attribute's value is valid.</p> <p>The API also assumes that the request will not contain any extraneous attribute (i.e., the request JSON will never contain any attribute that is not listed above).</p>
Failure	401 Unauthorized	If there is any kind of issue with the provided JWT a 401 error code is returned. Some examples are the JWT

		timing out, the authorization header being missing, the header being invalid, or any other kind of validity issue with the provided JWT necessary for authentication/authorization to access the protected entity Boats.
Failure	405 Method Not Allowed	The server generates a 405 error code whenever a method on an endpoint is requested that isn't supported. The server will generate an html error page and set the Allow header with the appropriate methods for the given endpoint.
Failure	406 Not Acceptable	If the user requests a media type the server can't offer via the Accept header then return 406 error status code. In this API the only acceptable media type the server will return is JSON. (Auto-generated HTML for errors may also occur)

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.
- When creating a new boat, loads will automatically be generated as an empty array ready to store loads.

Success

Status: 201 Created

```
{
  "id": 123,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "loads": [],
  "owner": "auth0|456a7f",
  "self": "https://final-project-walkerm7.uw.r.appspot.com/boats/123"
}
```

Failure: 400

Status: 400 Bad Request

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

Failure: 401

Status: 401 Unauthorized

```
{
  "code": "no auth header"
  "description": "Authorization header is missing"
}
{
  "code": "invalid_header"
```

```
"description": "Invalid header. Use an RS256 signed JWT Access Token"
}
{
  "code": "token_expired"
  "description": "token is expired"
}
{
  "code": "invalid_claims"
  "description": "incorrect claims, please check the audience and issuer"
}
{
  "code": "invalid_header"
  "description": "Unable to parse authentication token."
}
{
  "code": "no_rsa_key"
  "description": "No RSA key in JWKS"
}
```

Failure: 405

Status: 405 Method Not Allowed

Failure: 406

Status: 406 Not Acceptable

List all Boats with Pagination

List all the boats with pagination for the user with the provided JWT.

GET /boats

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	If there is any kind of issue with the provided JWT a 401 error code is returned. Some examples are the JWT timing out, the authorization header being missing, the header being invalid, or any other kind of validity issue with the provided JWT necessary for authentication/authorization to access the protected entity Boats.
Failure	405 Method Not Allowed	The server generates a 405 error code whenever a method on an endpoint is requested that isn't supported. The server will generate an html error page and set the Allow header with the appropriate methods for the given endpoint.
Failure	406 Not Acceptable	If the user requests a media type the server can't offer via the Accept header then return 406 error status code. In this API the only acceptable media type the server will return is JSON. (Auto-generated HTML for errors may also occur)

Response Examples

- For getting all boats with pagination the API fetches a certain limit of boats per page (5 in this case) and puts it into an array of objects called "boats". If and only if there are more boats than the ones given on a page, the "next" URL is generated that calculates the offset to get the next page of boats.
- The implementation of the API will also supply the total number of Boats that is associated with the user (even if on page 2 can always see the total number belonging to the user).
- It is also worth noting that if there were no boats this would return an empty array named "boats" that indicates there are currently no existing boats.

Success

Status: 200 OK

```
{
  "boats": [
    {
      "id": 123,
      "name": "Sea Witch",
      "type": "Catamaran",
      "length": 28,
      "loads": [
        {
          "id": 647,
          "self": "https://final-project-walkerm7.uw.r.appspot.com/loads/647"
        }
      ],
      "self": "https://final-project-walkerm7.uw.r.appspot.com/boats/123",
      "owner": "auth0|456a7f",
    },
    {
      "id": 456,
      "name": "Adventure",
      "type": "Yacht",
      "length": 50,
      "loads": [],
      "self": "https://final-project-walkerm7.uw.r.appspot.com/boats/456",
      "owner": "auth0|456a7f"
    },
    {
      "id": 789,
      "name": "Hocus Pocus",
      "type": "Sailboat",
      "length": 100,
      "loads": [],
      "self": "https://final-project-walkerm7.uw.r.appspot.com/boats/789",
      "owner": "auth0|456a7f"
    }
  ],
  "next": "https://final-project-walkerm7.uw.r.appspot.com/boats?limit=5&offset=5",
  "total_boats": 6
}
```

Failure: 401

Status: 401 Unauthorized

```
{
  "code": "no auth header"
  "description": "Authorization header is missing"
}
{
  "code": "invalid_header"
  "description": "Invalid header. Use an RS256 signed JWT Access Token"
}
```



```
{
  "code": "token_expired"
  "description": "token is expired"
}
{
  "code": "invalid_claims"
  "description": "incorrect claims, please check the audience and issuer"
}
{
  "code": "invalid_header"
  "description": "Unable to parse authentication token."
}
{
  "code": "no_rsa_key"
  "description": "No RSA key in JWKS"
}
```

Failure: 405

Status: 405 Method Not Allowed

Failure: 406

Status: 406 Not Acceptable

Get a Boat

Allows you to get an existing boat provided the proper JWT is provided as a boat is a protected entity.

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	401 Unauthorized	If there is any kind of issue with the provided JWT a 401 error code is returned. Some examples are the JWT timing out, the authorization header being missing, the header being invalid, or any other kind of validity issue with the provided JWT necessary for authentication/authorization to access the protected entity Boats.
Failure	403 Forbidden	When a user has a valid JWT but attempts to access a protected entity that belongs to another user a 403 error code is generated along with an error message stating that only a valid user can interact with the protected entity.
Failure	404 Not Found	No boat with this boat_id exists
Failure	405 Method Not Allowed	The server generates a 405 error code whenever a method on an endpoint is requested that isn't supported. The server will generate an html error page and set the Allow header with the appropriate methods for the given endpoint.
Failure	406 Not Acceptable	If the user requests a media type the server can't offer via the Accept header then return 406 error status code. In this API the only acceptable media type the server will return is JSON. (Auto-generated HTML for errors may also occur)

Response Examples

- Note that self is a URL that points to a resource's most direct location. It is generated on the fly by the API and is not stored at any time in the database (data store).

- Also note that the self URL starts with the {base URL} and then the address to the resource. In this case the base URL is: “https://final-project-walkerm7.uw.r.appspot.com” however it can be changed to anything that the API can host and has access to.

Success

Status: 200 OK

```
{
  "id": 123,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "loads": [],
  "self": "https://final-project-walkerm7.uw.r.appspot.com/boats/:boat_id",
}
```

Failure: 401

Status: 401 Unauthorized

```
{
  "code": "no_auth_header"
  "description": "Authorization header is missing"
}
{
  "code": "invalid_header"
  "description": "Invalid header. Use an RS256 signed JWT Access Token"
}
{
  "code": "token_expired"
  "description": "token is expired"
}
{
  "code": "invalid_claims"
  "description": "incorrect claims, please check the audience and issuer"
}
{
  "code": "invalid_header"
  "description": "Unable to parse authentication token."
}
{
  "code": "no_rsa_key"
  "description": "No RSA key in JWKS"
}
```

Failure: 403

Status: 403 Forbidden

```
{
  "Error": "This entity (boats) belongs to someone else and can only be retrieved by that owner"
}
```

```
}
```

Failure: 404

Status: 404 Not Found

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

Failure: 405

Status: 405 Method Not Allowed

Failure: 406

Status: 406 Not Acceptable

Delete a Boat

Allows you to delete a boat. Note that if there are any loads currently on the boat it will first go and unlink (unload) each load with the given boat before deletion. As this is a protected entity a valid JWT must be used for the user who owns the specific boat otherwise no access will be given.

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	401 Unauthorized	If there is any kind of issue with the provided JWT a 401 error code is returned. Some examples are the JWT timing out, the authorization header being missing, the header being invalid, or any other kind of validity issue with the provided JWT necessary for authentication/authorization to access the protected entity Boats.
Failure	403 Forbidden	When a user has a valid JWT but attempts to access a protected entity that belongs to another user a 403 error code is generated along with an error message stating that only a valid user can interact with the protected entity.
Failure	404 Not Found	No boat with this boat_id exists
Failure	405 Method Not Allowed	The server generates a 405 error code whenever a method on an endpoint is requested that isn't supported. The server will generate an html error page and set the Allow header with the appropriate methods for the given endpoint.
Failure	406 Not Acceptable	If the user requests a media type the server can't offer via the Accept header then return 406 error status code. In this API the only acceptable media type the server will return is JSON. (Auto-generated HTML for errors may also

		occur)
--	--	--------

Response Examples

Success

Status: 204 No Content

Failure: 401

Status: 401 Unauthorized

```
{
  "code": "no_auth_header"
  "description": "Authorization header is missing"
}
{
  "code": "invalid_header"
  "description": "Invalid header. Use an RS256 signed JWT Access Token"
}
{
  "code": "token_expired"
  "description": "token is expired"
}
{
  "code": "invalid_claims"
  "description": "incorrect claims, please check the audience and issuer"
}
{
  "code": "invalid_header"
  "description": "Unable to parse authentication token."
}
{
  "code": "no_rsa_key"
  "description": "No RSA key in JWKS"
}
```

Failure: 403

Status: 403 Forbidden

```
{
  "Error": "This entity (boats) belongs to someone else and can only be deleted by that owner"
}
```

Failure: 404

Status: 404 Not Found

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

Failure: 405

Status: 405 Method Not Allowed

Failure: 406

Status: 406 Not Acceptable

Change a Boat (PUT)

Allows you to create a new boat. A boat is a protected endpoint so a valid JWT must be provided.

PUT /boats:boat_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

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	<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>For this use case no validation of provided values for the attributes of the request is performed. It is assumed that if the request contains any of the listed attributes, then that attribute's value is valid.</p> <p>The API also assumes that the request will not contain any extraneous attribute (i.e., the request JSON will never contain any attribute that is not listed above).</p>
Failure	401 Unauthorized	If there is any kind of issue with the provided JWT a 401 error code is returned. Some examples are the JWT

		timing out, the authorization header being missing, the header being invalid, or any other kind of validity issue with the provided JWT necessary for authentication/authorization to access the protected entity Boats.
Failure	403 Forbidden	When a user has a valid JWT but attempts to access a protected entity that belongs to another user a 403 error code is generated along with an error message stating that only a valid user can interact with the protected entity.
Failure	404 Not Found	No boat with this boat_id exists
Failure	405 Method Not Allowed	The server generates a 405 error code whenever a method on an endpoint is requested that isn't supported. The server will generate an html error page and set the Allow header with the appropriate methods for the given endpoint.
Failure	406 Not Acceptable	If the user requests a media type the server can't offer via the Accept header then return 406 error status code. In this API the only acceptable media type the server will return is JSON. (Auto-generated HTML for errors may also occur)

Response Examples

Success

Status: 201 Created

```
{
  "id": 123,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "loads": []
  "owner": "auth0|456a7f",
  "self": "https://final-project-walkerm7.uw.r.appspot.com/boats/123"
}
```

Failure: 400

Status: 400 Bad Request

```
{
  "Error": "The request object has either an improper amount of attributes or the wrong attributes"
}
```

Failure: 401

Status: 401 Unauthorized

```
{
  "code": "no auth header"
  "description": "Authorization header is missing"
}
```



```
{
  "code": "invalid_header"
  "description": "Invalid header. Use an RS256 signed JWT Access Token"
}
{
  "code": "token_expired"
  "description": "token is expired"
}
{
  "code": "invalid_claims"
  "description": "incorrect claims, please check the audience and issuer"
}
{
  "code": "invalid_header"
  "description": "Unable to parse authentication token."
}
{
  "code": "no_rsa_key"
  "description": "No RSA key in JWKS"
}
```

Failure: 403

Status: 403 Forbidden

```
{
  "Error": "This entity (boats) belongs to someone else and can only be edited by
that owner"
}
```

Failure: 404

Status: 404 Not Found

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

Failure: 405

Status: 405 Method Not Allowed

Failure: 406

Status: 406 Not Acceptable

Update a Boat (PATCH)

Allows you to change any and all of the attributes in an existing boat that are used to create a new boat. A boat is a protected endpoint so a valid JWT must be provided.

PATCH /boats:boat_id

Request

Path Parameters

Name	Description
boat_id	ID of the boat

Request Body

Required

Request Body Format

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 Example

```
{
  "type": "Ghost Ship",
  "length": 28
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 Ok	
Failure	401 Unauthorized	If there is any kind of issue with the provided JWT a 401 error code is returned. Some examples are the JWT timing out, the authorization header being missing, the header being invalid, or any other kind of validity issue with the provided JWT necessary for authentication/authorization to access the protected entity Boats.
Failure	403 Forbidden	When a user has a valid JWT but attempts to access a protected entity that belongs to another user a 403 error code is generated along with an error message stating that only a valid user can interact with the protected entity.
Failure	404 Not Found	No boat with this boat_id exists

Failure	405 Method Not Allowed	The server generates a 405 error code whenever a method on an endpoint is requested that isn't supported. The server will generate an html error page and set the Allow header with the appropriate methods for the given endpoint.
Failure	406 Not Acceptable	If the user requests a media type the server can't offer via the Accept header then return 406 error status code. In this API the only acceptable media type the server will return is JSON. (Auto-generated HTML for errors may also occur)

Response Examples

Success

Status: 201 Created

```
{
  "id": 123,
  "name": "Sea Witch",
  "type": "Ghost Ship",
  "length": 28,
  "loads": [],
  "owner": "auth0|456a7f",
  "self": "https://final-project-walkerm7.uw.r.appspot.com/boats/123"
}
```

Failure: 401

Status: 401 Unauthorized

```
{
  "code": "no_auth_header"
  "description": "Authorization header is missing"
}
{
  "code": "invalid_header"
  "description": "Invalid header. Use an RS256 signed JWT Access Token"
}
{
  "code": "token_expired"
  "description": "token is expired"
}
{
  "code": "invalid_claims"
  "description": "incorrect claims, please check the audience and issuer"
}
{
  "code": "invalid_header"
  "description": "Unable to parse authentication token."
}
{
  "code": "no_rsa_key"
  "description": "No RSA key in JWKS"
}
```

```
}
```

Failure: 403

Status: 403 Forbidden

```
{  
"Error": "This entity (boats) belongs to someone else and can only be edited by  
that owner"  
}
```

Failure: 404

Status: 404 Not Found

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

Failure: 405

Status: 405 Method Not Allowed

Failure: 406

Status: 406 Not Acceptable

Load Assigned to Boat

A load is assigned (loaded onto) a boat. Although a boat is a protected resource loading and unloading is an unprotected action on a boat and does not require a JWT/authentication.

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 is not stored on another boat already.
Failure	403 Forbidden	The load is already loaded on another boat so it is unable to be assigned to this boat.
Failure	404 Not Found	The specified boat and/or load does not exist
Failure	405 Method Not Allowed	The server generates a 405 error code whenever a method on an endpoint is requested that isn't supported. The server will generate an html error page and set the Allow header with the appropriate methods for the given endpoint.

Response Examples

Success

Status: 204 No Content

Failure: 403

Status: 403 Forbidden

```
{
  "Error": "The load is already loaded on another boat"
}
```

Failure: 404

Status: 404 Not Found

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

Failure: 405

Status: 405 Method Not Allowed

Comment

- A boat can have multiple loads.
- A load can only ever be on one boat.
- The API must test/make sure that the load isn't already loaded onto another boat before being assigned to the target boat

Unassign/Remove a Load from a Boat

A load is removed from the target boat. The boat can have more loads or now be empty. The load now has no carrier and can be assigned to any boat. Again though a boat is protected, loading/unloading is purposefully not protected.

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 assigned to 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 assigned to the boat with this boat_id.
Failure	405 Method Not Allowed	The server generates a 405 error code whenever a method on an endpoint is requested that isn't supported. The server will generate an html error page and set the Allow header with the appropriate methods for the given endpoint.

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"
}
```

Status: 405 Method Not Allowed

Create a Load

Allows you to create a new load. (loads are not protected)

POST /loads

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
item	The name of the item that the load is carrying.	Yes
volume	The volume of the load.	Yes
color	The color of the load	Yes

Request Body Example

```
{
  "volume": 5,
  "color": "red",
  "item": "Lego Blocks"
}
```

Response

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 required attributes, the load must not be created, and 400 status code must be returned.</p> <p>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.</p> <p>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).</p>
Failure	405 Method Not Allowed	<p>The server generates a 405 error code whenever a method on an endpoint is requested that isn't supported. The server will generate an html error page and set the</p>

		Allow header with the appropriate methods for the given endpoint.
Failure	406 Not Acceptable	If the user requests a media type the server can't offer via the Accept header then return 406 error status code. In this API the only acceptable media type the server will return is JSON. (Auto-generated HTML for errors may also occur)

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 object containing the ID of the boat this load is currently on. If the load is not on a boat then, the value of `carrier` should be null. This will always be the case for when a load is first created.

Success

Status: 201 Created

```
{
  "id": 123,
  "volume": 5,
  "carrier": null,
  "color": "red",
  "item": "Lego Blocks",
  "creation_date": "10/20/2023"
  "self": "https://final-project-walkerm7.uw.r.appspot.com/loads/123",
}
```

Failure

Status: 400 Bad Request

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

Failure: 405

Status: 405 Method Not Allowed

Failure: 406

Status: 406 Not Acceptable

List all Loads with Pagination

List all the loads with pagination.

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

- This is implemented the same way as viewing all boats (even uses the same abstracted function in API).

Success

Status: 200 OK

```
{
  "loads": [
    {
      "id": 123,
      "volume": 5,
      "carrier": {
        "id": 346,
        "self": "https://final-project-walkerm7.uw.r.appspot.com/boats/346"
      },
      "color": "pink",
      "item": "Lego Blocks",
      "creation_date": "10/20/2023"
    },
    {
      "id": 412,
      "volume": 3,
      "carrier": {
        "id": 456,
        "self": "https://final-project-walkerm7.uw.r.appspot.com/boats/456"
      },
      "color": "blue",
      "item": "Food",
      "creation_date": "10/25/2023"
    }
  ]
}
```

```
    "self": "https://final-project-walkerm7.uw.r.appspot.com/loads/412",
  }
  {
    "id": 543,
    "volume": 10,
    "carrier": {
      "id": 746,
      "self": "https://final-project-walkerm7.uw.r.appspot.com/boats/746"
    },
    "color": "white",
    "item": "Stuffed Animals",
    "creation_date": "10/26/2023"
    "self": "https://final-project-walkerm7.uw.r.appspot.com/loads/543",
  }
  ...
],
"next": "https://final-project-walkerm7.uw.r.appspot.com/loads?limit=5&offset=5"
"total_loads": 7
}
```

Get 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
Failure	405 Method Not Allowed	The server generates a 405 error code whenever a method on an endpoint is requested that isn't supported. The server will generate an html error page and set the Allow header with the appropriate methods for the given endpoint.
Failure	406 Not Acceptable	If the user requests a media type the server can't offer via the Accept header then return 406 error status code. In this API the only acceptable media type the server will return is JSON. (Auto-generated HTML for errors may also occur)

Response Examples

- Again note that the "self" attributes are not stored in database but rather generated on the fly via the API.

Success

Status: 200 OK
<pre>{ "id": 123, "volume": 5, "carrier": { "id": 346, "self": "https://final-project-walkerm7.uw.r.appspot.com/boats/346" }, "color": "orange", "item": "Lego Blocks", "creation_date": "10/20/2023" "self": "https://final-project-walkerm7.uw.r.appspot.com/loads/123",</pre>

```
}
```

Failure

Status: 404 Not Found

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

Failure: 405

Status: 405 Method Not Allowed

Failure: 406

Status: 406 Not Acceptable

Change a Load (PUT)

Allows you to change everything about a load that currently exists. (All the attributes that are required to make a new load)

PUT /loads:load_id

Request

Path Parameters

Name	Description
load_id	ID of the load

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
item	The name of the item that the load is carrying.	Yes
volume	The volume of the load.	Yes
color	The color of the load	Yes

Request Body Example

```
{
  "volume": 10,
  "color": "blue",
  "item": "Bottled water",
}
```

Response

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 4 required attributes, the load must not be created, and 400 status code must be returned.</p> <p>For this use case no validation of provided values for the attributes of the request is performed. It is assumed that if the request contains any of the listed attributes, then that attribute's value is valid.</p> <p>The API also assumes that the request will not contain any extraneous attribute (i.e., the request JSON will never contain any attribute that is not listed above).</p>

Failure	404 Not Found	No load with this load_id exists
Failure	405 Method Not Allowed	The server generates a 405 error code whenever a method on an endpoint is requested that isn't supported. The server will generate an html error page and set the Allow header with the appropriate methods for the given endpoint.
Failure	406 Not Acceptable	If the user requests a media type the server can't offer via the Accept header then return 406 error status code. In this API the only acceptable media type the server will return is JSON. (Auto-generated HTML for errors may also occur)

Response Examples

Success

Status: 201 Created

```
{
  "id": 123,
  "volume": 10,
  "carrier": {
    "id": 346,
    "self": "https://final-project-walkerm7.uw.r.appspot.com/boats/346"
  },
  "color": "blue",
  "item": "Bottled water",
  "creation_date": "10/20/2023"
  "self": "https://final-project-walkerm7.uw.r.appspot.com/loads/123",
}
```

Failure: 400

Status: 400 Bad Request

```
{
  "Error": "The request object has either an improper amount of attributes or the wrong attributes"
}
```

Failure: 404

Status: 404 Not Found

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

Failure: 405

Status: 405 Method Not Allowed

Failure: 406

Status: 406 Not Acceptable

Update a Load (PATCH)

Allows you to change any number of attributes in a load that is used to create a new load (the attributes that aren't automatically generated or have to be linked).

PATCH /loads:load_id

Request

Path Parameters

Name	Description
load_id	ID of the load

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Description	Required?
item	The name of the item that the load is carrying.	No
volume	The volume of the load.	No
color	The color of the load	No

Request Body Example

```
{
  "item": "Anime"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	404 Not Found	No load with this load_id exists
Failure	405 Method Not Allowed	The server generates a 405 error code whenever a method on an endpoint is requested that isn't supported. The server will generate an html error page and set the Allow header with the appropriate methods for the given endpoint.
Failure	406 Not Acceptable	If the user requests a media type the server can't offer via the Accept header then return 406 error status code. In this API the only acceptable media type the server will return is JSON. (Auto-generated HTML for errors may also occur)

Response Examples

Success

Status: 201 Created


```
{
  "id": 123,
  "volume": 10,
  "carrier": {
    "id": 346,
    "self": "https://final-project-walkerm7.uw.r.appspot.com/boats/346"
  },
  "color": "blue",
  "item": "Anime",
  "creation_date": "10/20/2023"
  "self": "https://final-project-walkerm7.uw.r.appspot.com/loads/123",
}
```

Failure: 404

Status: 404 Not Found

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

Failure: 405

Status: 405 Method Not Allowed

Failure: 406

Status: 406 Not Acceptable

Delete a Load

Allows you to delete a load. If the load has a carrier (is on a boat) then the load is removed from that boat before deleting the 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": "No load with this load_id exists"
}
```

Get Users

List all the users who have registered and received a JWT for the first time (not paginated and doesn't keep count of total entities returned).

GET /users

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	405 Method Not Allowed	The server generates a 405 error code whenever a method on an endpoint is requested that isn't supported. The server will generate an html error page and set the Allow header with the appropriate methods for the given endpoint.
Failure	406 Not Acceptable	If the user requests a media type the server can't offer via the Accept header then return 406 error status code. In this API the only acceptable media type the server will return is JSON. (Auto-generated HTML for errors may also occur)

Response Examples

Success

```
Status: 200 OK
[ {
  "id": "auth0|456a7f",
  "email": "fake@email.com"
},
{
  "id": "auth0|84eab3",
  "email": "silly@email.com"
} ]
```

Failure: 405

Status: 405 Method Not Allowed

Failure: 406

Status: 406 Not Acceptable