Portfolio Project: API Spec

CS 493: Cloud Application Development

Fall 2023

Oregon State University

Last Update: Dec 9. 11:30 am Pacific

**GAE URL:** [**https://cs493-protfolio-lottotlt.uc.r.appspot.com**](https://cs493-protfolio-lottotlt.uc.r.appspot.com)

**Content**

[**Change Log**](#_Toc153120403)

[**Data Model**](#_Toc153120404)

[**Create User (Method 1 – Postman)**](#_Toc153120408)

[**Create User (Method 2 – Webpage)**](#_Toc153120422)

[**Login User (Method 1 – Postman)**](#_Toc153120423)

[**Login User (Method 2 – Webpage)**](#_Toc153120436)

[**Get all Users**](#_Toc153120437)

[**Create a Boat (Required user JWT)**](#_Toc153120446)

[**Get all Boats (Optional user JWT)**](#_Toc153120460)

[**Get a Boat (Required user JWT)**](#_Toc153120471)

[**Patch a Boat (Required user JWT)**](#_Toc153120483)

[**Put a Boat (Required user JWT)**](#_Toc153120497)

[**Delete a Boat (Required user JWT)**](#_Toc153120511)

[**Create a Load**](#_Toc153120522)

[**Get all Loads**](#_Toc153120535)

[**Get a Load**](#_Toc153120544)

[**Patch a Load**](#_Toc153120554)

[**Put a Load**](#_Toc153120567)

[**Delete a Load**](#_Toc153120580)

[**Managing Loads - Assign Load to a Boat (Required user JWT)**](#_Toc153120590)

[**Managing Loads - Remove Load from a Boat (Required user JWT)**](#_Toc153120602)

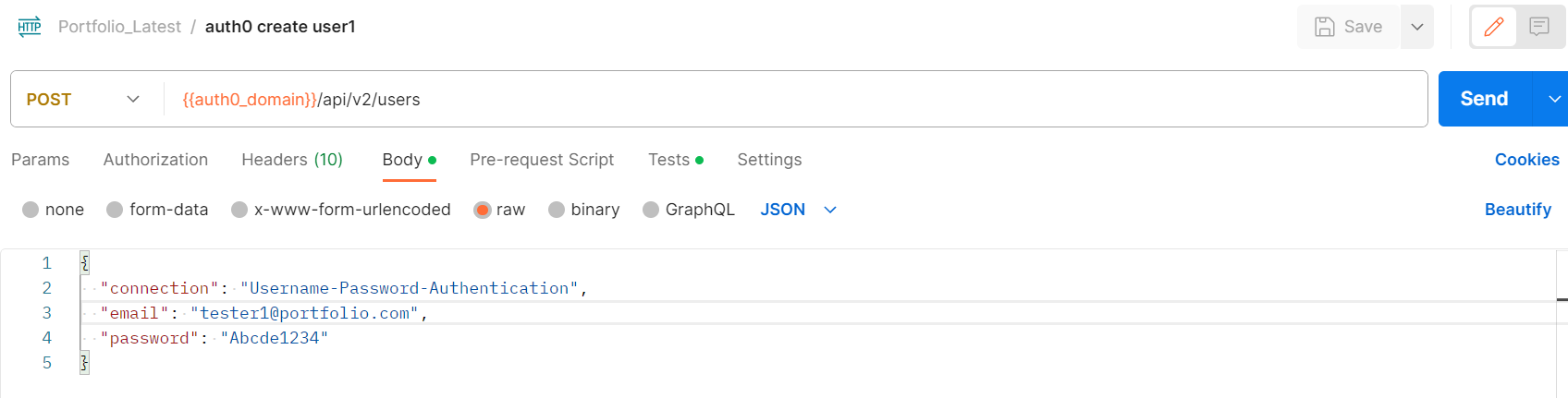
[**Miscellaneous – Unallowed HTTP verb for routes not stated above**](#_Toc153120613)

# Change Log

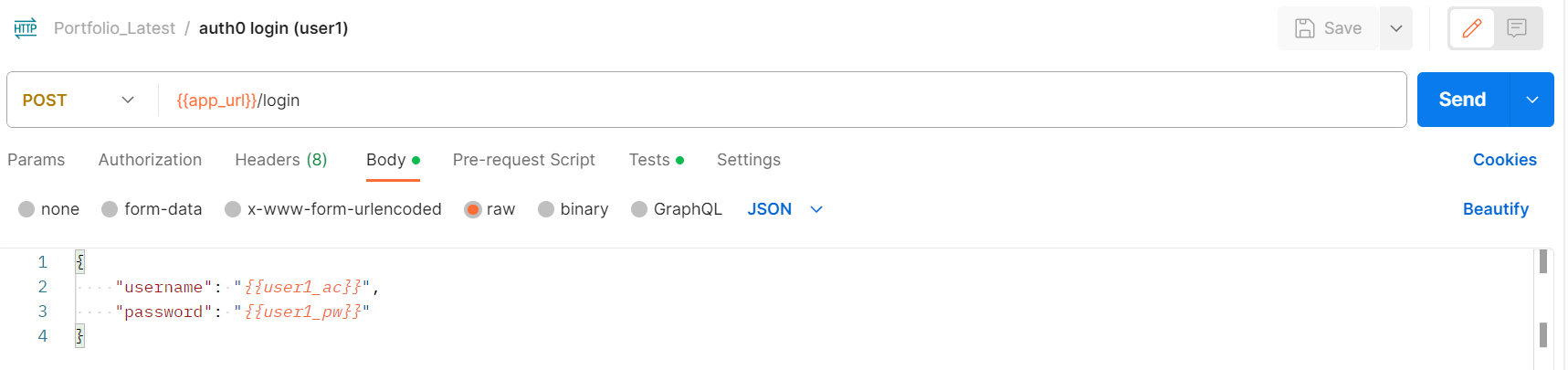
|  |  |  |
| --- | --- | --- |
| **Version** | **Change** | **Date** |
| 1.0 | Initial version. | Dec 9, 2023 |

# Remarks

* Please follow the sequence of the test cases in Postman.
* The project will be submitted with no records in Datastore and Auth0.
* In Postman, the default username as:
  + [tester1@portfolio.com](mailto:tester1@portfolio.com)
  + [tester2@portfolio.com](mailto:tester2@portfolio.com)
* **For any other username or password, please enter in the body of the request in Postman.**



* **For any users registered on the Webpage, please substitute the {{user\_ac}} and {{user\_pc}} and send the login request in Postman** for implementing the following test cases in Postman.



* This App will **use the content of [‘sub’] from Auth0 to be the unique identifier for User** (user\_id as stored in Datastore)
* **Every boat must have an owner** (user JWT is required to do the CRUD operations related to boat), while load can be created from everyone.
* For Postman variable **user\_id1, user\_id2, jwt1, jwt2, these variables will be generated on the fly on the test cases.**

# Data Model

The app stores **THREE** kinds of entities in Datastore, **Boats**, **Loads** and **Users**.

## Boats (Require user JWT)

|  |  |  |
| --- | --- | --- |
| **Property** | **Data Type** | **Notes** |
| id | Integer | The id of the boat. Datastore automatically generates it. |
| name | String | The name of the boat. |
| type | String | The type of the Boat. E.g., Sailboat, Catamaran, etc. |
| length | Integer | Length of the Boat in feet. |
| loads | Array of objects | **Empty array as default**. Load(s) - id and self-link will be added to the boat. A boat can have many loads, but a load can only be carried by one boat. |
| owner | String | The id of the user. Only the owner of the boat can perform the CRUD operations for the boat. |
| self | String | **Not a property stored in Datastore**, but will be displayed the URL of the resource when having the response body returned. |

## Loads

|  |  |  |
| --- | --- | --- |
| **Property** | **Data Type** | **Notes** |
| id | Integer | The id of the load. Datastore automatically generates it. |
| volume | Integer | The volume of the load. |
| carrier | Object | **Null as default.** Boat – id, name and self will be added to the load. A load can only be carried by one boat. |
| item | String | The name / content of the load. |
| creation\_date | String | The date that the load is created. |
| self | String | **Not a property stored in Datastore**, but will be displayed the URL of the resource when having the response body returned. |

## Users

|  |  |  |
| --- | --- | --- |
| **Property** | **Data Type** | **Notes** |
| id | Integer | The id of the load. Datastore automatically generates it. |
| username | String | The username of the user (e.g. email address). |
| user\_id | String | The unique identifier of the user from Auth0. |

# Create User (Method 1 – Postman)

Allows you to create a new user.

|  |
| --- |
| POST {{DOMAIN}}/api/v2/users |

## Authorization

Bearer Token (Management API Access Tokens)

* In Postman, the token is provided by developer
* Token expires in a month

## Request

### Path Parameters

None

### Request Body

Required

**Default in Postman:** [**tester1@portfolio.com**](mailto:tester1@portfolio.com) **and** [**tester2@portfolio.com**](mailto:tester2@portfolio.com)

### Request Body Format

JSON

### Request JSON Attributes

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Required?** |
| connection | Connection type to the database | Required |
| email | User login account | Required |
| password | User login password | Required |

### Request Body Example

|  |
| --- |
| {    "connection": "Username-Password-Authentication",    "email": "tester1@portfolio.com",    "password": "Abcde1234"  } |

## Response

### Response Body Format

JSON

### Response Statuses

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status Code** | **Notes** |
| Success | 201 Created |  |
| Failure | 400 Bad Request | Return 400 status code if:   * connection type is not valid * password is too weak * email not in proper format |
| Failure | 401 Unauthorized | Return 401 status code if:   * no bearer token is provided. |
| Failure | 409 Conflict | Return 409 status code if:   * the email is already registered. |

### Response Examples

#### Success

|  |
| --- |
| Status: 201 Created  {      "created\_at": "2023-12-10T08:17:43.454Z",      "email": "tester5@portfolio.com",      "email\_verified": **false**,      "identities": [          {              "connection": "Username-Password-Authentication",              "user\_id": "65757427e94488fb5f77f5d6",              "provider": "auth0",              "isSocial": **false**          }      ],      "name": "tester5@portfolio.com",      "nickname": "tester5",      "picture": "https://s.gravatar.com/avatar/550127caedcceddfe9fd2e9085330c29?s=480&r=pg&d=https%3A%2F%2Fcdn.auth0.com%2Favatars%2Fte.png",      "updated\_at": "2023-12-10T08:17:43.454Z",      "user\_id": "auth0|65757427e94488fb5f77f5d6"  } |

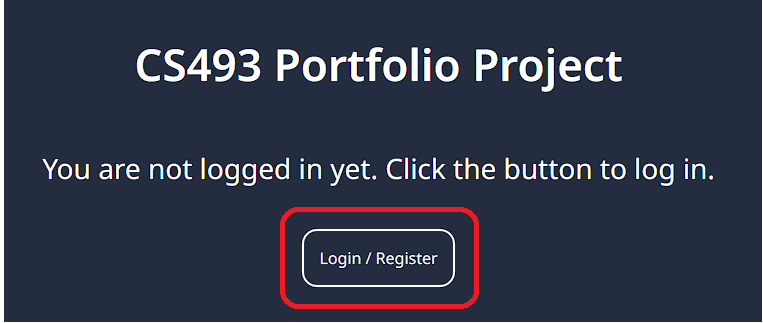
#### Failure

|  |
| --- |
| Status: 400 Bad Request  {      "statusCode": 400,      "error": "Bad Request",      "message": "PasswordStrengthError: Password is too weak"  }  Status: 400 Bad Request  {      "statusCode": 400,      "error": "Bad Request",      "message": "The connection does not exist.",      "errorCode": "inexistent\_connection"  }  Status: 400 Bad Request  {      "statusCode": 400,      "error": "Bad Request",      "message": "Payload validation error: 'Object didn't pass validation for format email: tester' on property email (The user's email).",      "errorCode": "invalid\_body"  }  Status: 401 Unauthorized  {      "statusCode": 401,      "error": "Unauthorized",      "message": "Missing authentication"  }  Status: 409 Conflict  {      "statusCode": 409,      "error": "Conflict",      "message": "The user already exists.",      "errorCode": "auth0\_idp\_error"  } |

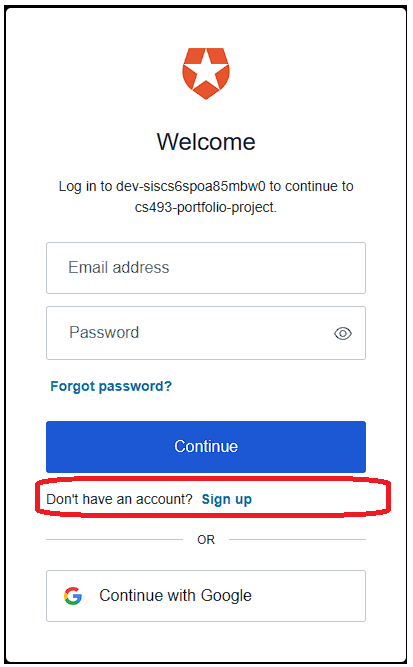
# Create User (Method 2 – Webpage)

Step 1:

Webpage URL: <https://cs493-protfolio-lottotlt.uc.r.appspot.com>

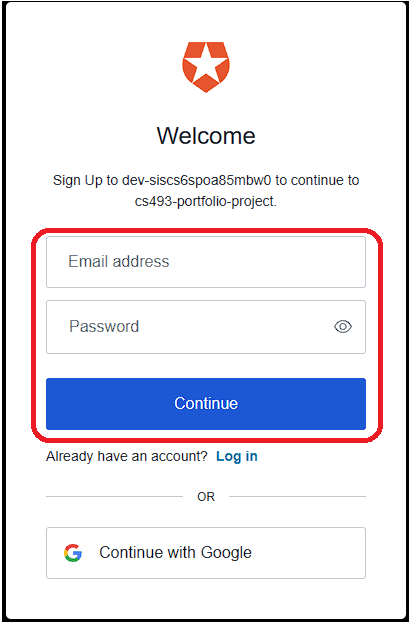


Step 2:



Step 3 (Enter email and password, then click “Continue”)

* Please follow the input requirements on the webpage.



# Login User (Method 1 – Postman)

Allows you to create a new user.

|  |
| --- |
| POST /login |

## Request

### Path Parameters

None

### Request Body

Required

### Request Body Format

JSON

### Request JSON Attributes

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Required?** |
| username | User login account | Required |
| password | User login password | Required |

### Request Body Example

|  |
| --- |
| {      "username": "*{{user1\_ac}}*",      "password": "*{{user1\_pw}}*"  } |

## Response

### Response Body Format

JSON

### Response Statuses

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status Code** | **Notes** |
| Success | 200 OK |  |
| Failure | 403 Forbidden | Return 403 status code if:   * invalid username or password. |

### Response Examples

#### Success

|  |
| --- |
| Status: 200 OK  {      "access\_token": "eyJhbGciOiJkaXIiLCJlbmMiOiJBMjU2R0NNIiwiaXNzIjoiaHR0cHM6Ly9kZXYtc2lzY3M2c3BvYTg1bWJ3MC51cy5hdXRoMC5jb20vIn0..YoSwS7kLe-WoPpfu.74pZJMFjbpgRrM9gy68iXFkj2gkTCUUIwmCpiPOjdpHRanYcJNl8XvAEX6X9xP3LRyJLojVPzam4cALbowmqnKYHvgDBSW4Hig8Q8ZgTBHijVtvcSgGKjEMIOPl23lVx8F34WJLRksdqtXdm31MU\_3v7uTk3oFtWjV0JKjXK3F10\_uX-6hDuJq9FfpJsooV41NZFsWDOSoNxmBwoV4J-bhlZ6i7grWJ4FPwXfHQ2A2pyNjT5owIALZRI264ksOfBrExlHef7w\_fyThfdK2JCoP2iePrDg-kSPX-GM7z9ghz0-Y1wneaBkk7X1m9Sf6OS0U2utARpnRFr172-cB1qLX6o0jETbdl4KNc4o98gkBZxtkuOGNPAwrirsmsMEF5KMg.GUu6gqFBAr7Y2pOrwOG58A",      "id\_token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6IlpLbHBzanpMRWlmOXZXaTd5VDN5NCJ9..Tiix86TszHmuXVPrs491Tko5S5me877eJz8KCpxTWNKwOtFj\_1Tta1-wYI01eaGI\_TcX5r1snZoyrZF6puS4r\_2qq3H3AGhv0yMoR2Ql5mOpv2nOaomx1FvAIlFzkfnrNeuzFHZ6In-0wDR0cmgQFDJUeC0mMAPoN6hHIpOjFrWblonQg92XU9GHHtl0a1llXXntfGAlw1KKfcwd-K3RGc5ZGw2w1CAQoxutPasQDZeGXTdadYJqgocaiwak0dm5JITSh\_1bMYqjOs5OjfDQtPdLcB5\_1x5m\_yOUFlF68lYHiNqNHbJxv4HyyTG\_4nw9NKIpyT4D3cYTEN2xPKBurg",      "scope": "openid profile email address phone",      "expires\_in": 86400,      "token\_type": "Bearer"  } |

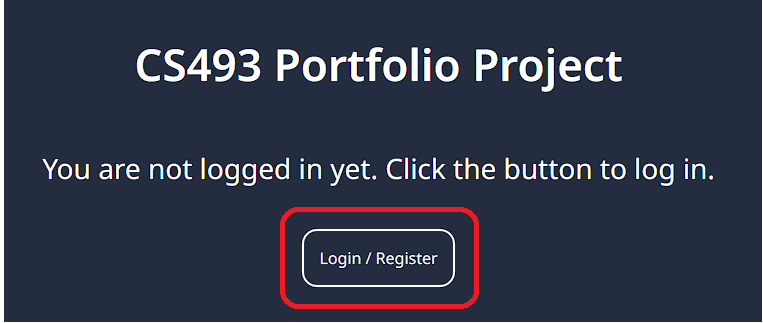
#### Failure

|  |
| --- |
| Status: 403 Forbidden  {      "error": "invalid\_grant",      "error\_description": "Wrong email or password."  } |

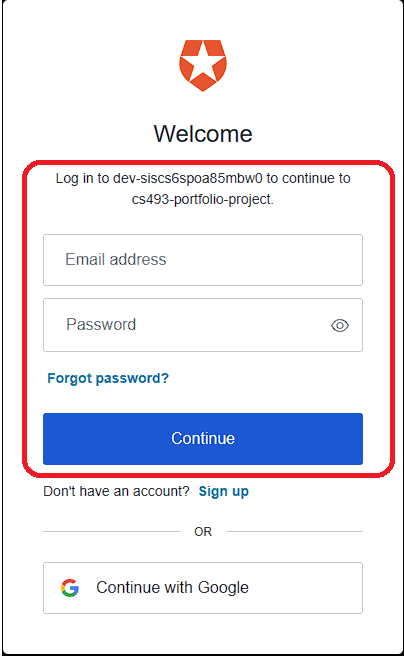
# Login User (Method 2 – Webpage)

Step 1:

Webpage URL: <https://cs493-protfolio-lottotlt.uc.r.appspot.com>

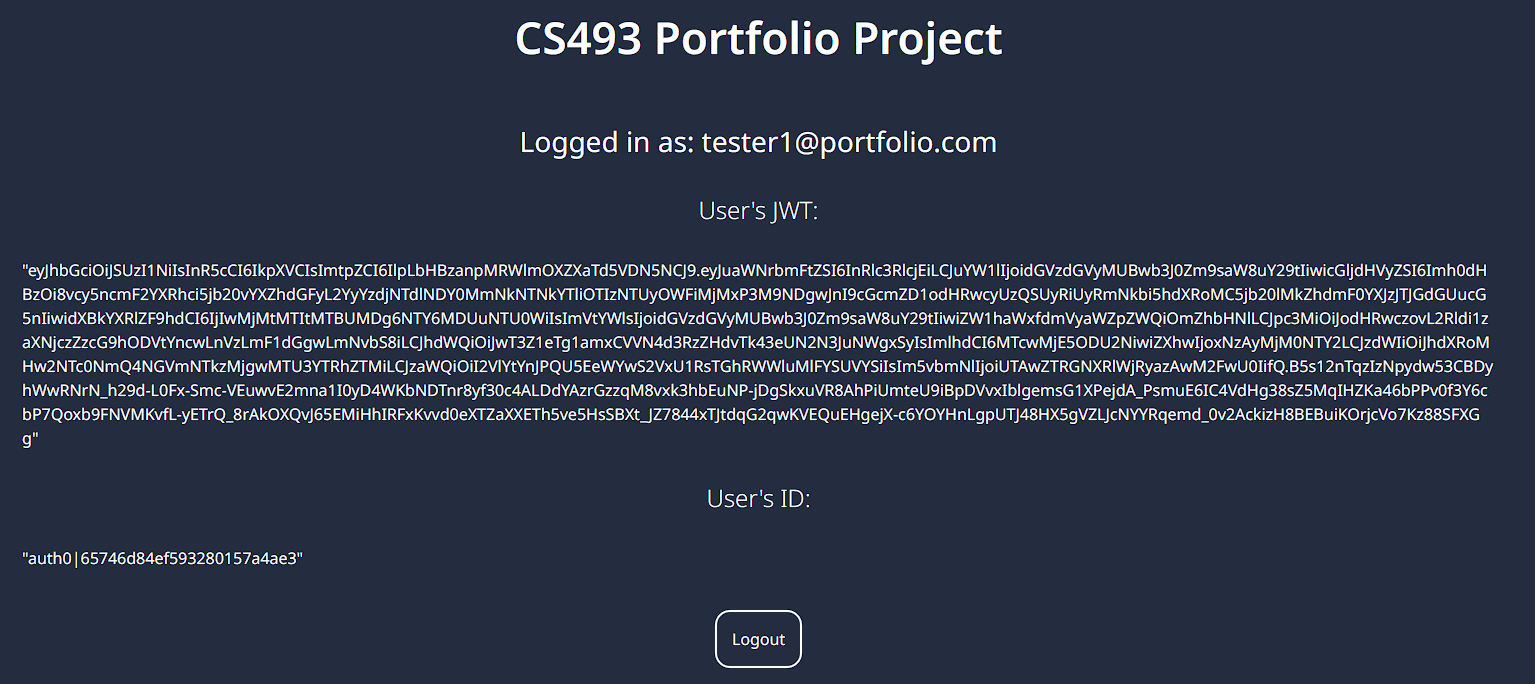


Step 2:



Step 3:

* After successfully register or login, the webpage will display the user’s JWT and unique ID.



# Get all Users

Allows you to get all registered users. If the user is registered through Postman, **the data will be recorded and be shown only after the 1st login of that account.**

|  |
| --- |
| 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

#### Success

|  |
| --- |
| Status: 200 OK  [      {          "user\_id": "auth0|6575778bef593280157b7eb7",          "username": "tester444@portfolio.com",          "id": 5124060585394176      },      {          "username": "tester3@portfolio.com",          "user\_id": "auth0|65746e829fa97e13efcd24dd",          "id": 5692744588591104      }  ] |

# Create a Boat (Required user JWT)

Allows you to create a new Boat.

|  |
| --- |
| POST /boats |

## Authorization

Bearer Token

## Request

### Path Parameters

None

### Request Body

Required

### Request Body Format

JSON

### Request JSON Attributes

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Required?** |
| name | The name of the boat. | Required |
| type | The type of the boat. E.g., Sailboat, Catamaran, etc. | Required |
| length | Length of the boat in feet. | Required |

### 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 | Return 400 status code if:   * missing one of the required attributes. |
| Failure | 401 Unauthorized | Return 401 status code if:   * no bearer token (user JWT) is provided. |
| Failure | 406 Not Acceptable | Return 406 status code if:   * the client request response type is not in application/json. |
| Failure | 415 Unsupported Media Type | Return 415 status code if:   * the request body is not in application/json. |

### Response Examples

* Datastore will automatically generate an ID and store it with the entity being created.
* The “self” attribute will not be stored in Datastore, but for displaying in the response body only.

#### Success

|  |
| --- |
| Status: 201 Created  {      "name": "Sea Witch",      "type": "Catamaran",      "length": 28,      "loads": [],      "owner": "auth0|65746d84ef593280157a4ae3",      "id": 5768473451954176,      "self": "https://cs493-protfolio lottotlt.uc.r.appspot.com/boats/5768473451954176"  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  {      "Error": "The request object is missing one of the required attributes."  }  Status: 401 Unauthorized  {      "Error": "Missing or invalid JWTs."  }  Status: 406 Not Acceptable  {      "Error": "Request MIME type is not acceptable."  }  Status: 415 Unsupported Media Type  {      "Error": "Server only accepts application/json data."  } |

# Get all Boats (Optional user JWT)

Allows you to get all boats from a particular user (with JWT is provided) or all boats in Datastore (without user JWT)

|  |
| --- |
| GET /boats |

## Authorization

Bearer token (Optional)

## Request

### Path Parameters

None

### Request Body

None

## Response

### Response Body Format

JSON

### Response Statuses

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status Code** | **Notes** |
| Success | 200 OK |  |

### Response Examples

* A maximum of 5 records will be fetched per request (for pagination). The “next” attribute will direct to the following 5 boats.
* The “self” attribute will not be stored in Datastore, but for displaying in the response body only.

#### Success

Case 1: with user1 JWT – List all user1’s boats

|  |
| --- |
| Status: 200 OK  {      "boats": [          {              "length": 15,              "type": "Yacht",              "name": "Lucifer",              "owner": "auth0|65746d84ef593280157a4ae3",              "loads": [],              "id": 5637013092958208,              "self": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/boats/5637013092958208"          },          {              "type": "Motor Boat",              "owner": "auth0|65746d84ef593280157a4ae3",              "loads": [],              "length": 25,              "name": "Abraham",              "id": 5679858948505600,              "self": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/boats/5679858948505600"          },          {              "loads": [],              "type": "Yacht",              "name": "401K",              "length": 18,              "owner": "auth0|65746d84ef593280157a4ae3",              "id": 5696855140728832,              "self": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/boats/5696855140728832"          },          {              "type": "Sailing Boat",              "length": 13,              "loads": [],              "owner": "auth0|65746d84ef593280157a4ae3",              "name": "Calypso",              "id": 5700115691995136,              "self": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/boats/5700115691995136"          },          {              "loads": [],              "length": 18,              "name": "Superman",              "owner": "auth0|65746d84ef593280157a4ae3",              "type": "Yacht",              "id": 5716170548183040,              "self": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/boats/5716170548183040"          }      ],      "next": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/boats?limit=5&offset=5"  } |

#### Success

Case 2: without JWT – List all recorded boats

|  |
| --- |
| Status: 200 OK  {      "boats": [          {              "type": "Yacht",              "name": "Lucifer",              "owner": "auth0|65746d84ef593280157a4ae3",              "loads": [],              "length": 15,              "id": 5637013092958208,              "self": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/boats/5637013092958208"          },          {              "owner": "auth0|65746d85e94488fb5f76c42e",              "length": 999,              "type": "hello world",              "loads": [],              "name": "user2 boat",              "id": 5674891550392320,              "self": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/boats/5674891550392320"          },          {              "length": 25,              "name": "Abraham",              "owner": "auth0|65746d84ef593280157a4ae3",              "type": "Motor Boat",              "loads": [],              "id": 5679858948505600,              "self": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/boats/5679858948505600"          },          {              "name": "401K",              "type": "Yacht",              "owner": "auth0|65746d84ef593280157a4ae3",              "loads": [],              "length": 18,              "id": 5696855140728832,              "self": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/boats/5696855140728832"          },          {              "length": 13,              "loads": [],              "name": "Calypso",              "owner": "auth0|65746d84ef593280157a4ae3",              "type": "Sailing Boat",              "id": 5700115691995136,              "self": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/boats/5700115691995136"          }      ],      "next": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/boats?limit=5&offset=5"  } |

Case 2 (Continue): without JWT – List all recorded boats (boats in “next”, with user2’s boats)

|  |
| --- |
| Status: 200 OK  {      "boats": [          {              "type": "Yacht",              "loads": [],              "name": "Superman",              "length": 18,              "owner": "auth0|65746d84ef593280157a4ae3",              "id": 5716170548183040,              "self": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/boats/5716170548183040"          },          {              "type": "Yacht",              "loads": [],              "owner": "auth0|65746d84ef593280157a4ae3",              "length": 18,              "name": "New World",              "id": 5761076981399552,              "self": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/boats/5761076981399552"          }      ]  } |

# Get a Boat (Required user JWT)

Allows you to get a particular boat.

|  |
| --- |
| GET /boats/:boat\_id |

## Authorization

Bearer token

## 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 | 400 Bad Request | Return 400 status code if:   * missing one of the required attributes. |
| Failure | 401 Unauthorized | Return 401 status code if:   * no bearer token (user JWT) is provided. |
| Failure | 403 Forbidden | Return 403 status code if:   * the boat belongs to other user. |
| Failure | 404 Not Found | Return 404 status code if:   * invalid boat\_id is provided. |
| Failure | 406 Not Acceptable | Return 406 status code if:   * the client request response type is not in application/json. |
| Failure | 415 Unsupported Media Type | Return 415 status code if:   * the request body is not in application/json. |

### Response Examples

* The “self” attribute will not be stored in Datastore, but for displaying in the response body only.

#### Success

Case 1: Without load

|  |
| --- |
| Status: 200 OK  {      "boats": [          {              "type": "Catamaran",              "length": 28,              "loads": [],              "owner": "auth0|65746d84ef593280157a4ae3",              "name": "Sea Witch",              "id": 5677209289228288,              "self": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/boats/5677209289228288"          }      ]  } |

#### Success

Case 2: With load

|  |
| --- |
| Status: 200 OK  {      "owner": "auth0|65746d84ef593280157a4ae3",      "length": 28,      "type": "Catamaran",      "loads": [          {              "self": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/loads/5747559041597440",              "id": 5747559041597440          }      ],      "name": "Sea Witch",      "id": 5677209289228288,      "self": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/boats/5677209289228288"  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  {      "Error": "The request object is missing one of the required attributes."  }  Status: 401 Unauthorized  {      "Error": "Missing or invalid JWTs."  }  Status: 403 Forbidden  {      "Error": "Valid JWT but the boat is owned by others."  }  Status: 404 Not Found  {      "Error": "No boat with this boat\_id exists."  }  Status: 406 Not Acceptable  {      "Error": "Request MIME type is not acceptable."  }  Status: 415 Unsupported Media Type  {      "Error": "Server only accepts application/json data."  } |

# 

# 

# 

# 

# 

# 

# 

# Patch a Boat (Required user JWT)

Allows you to edit 1 or more attribute(s) of a particular boat. Cannot manually modify “loads” in this route.

|  |
| --- |
| PATCH /boats/:boat\_id |

## Authorization

Bearer token

## 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. | Optional |
| type | The type of the boat. E.g., Sailboat, Catamaran, etc. | Optional |
| length | Length of the boat in feet. | Optional |

### Request Body Example

|  |
| --- |
| {      "name": "Sea Witch - Modified"  } |

## Response

### Response Body Format

JSON

### Response Statuses

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status Code** | **Notes** |
| Success | 200 OK | Return 200 status code if (pass all):   * only the allowed attribute(s) in the request body (except loads) * application/json for request body and return MIME type. * valid user JWT is provided. |
| Failure | 400 Bad Request | Return 400 status code if:   * invalid change of “loads” in this route. * Invalid attribute(s) is found on the request body. |
| Failure | 401 Unauthorized | Return 401 status code if:   * no bearer token (user JWT) is provided. |
| Failure | 403 Forbidden | Return 403 status code if:   * the boat belongs to other user. |
| Failure | 404 Not Found | Return 404 status code if:   * invalid boat\_id is provided. |
| Failure | 406 Not Acceptable | Return 406 status code if:   * the client request response type is not in application/json. |
| Failure | 415 Unsupported Media Type | Return 415 status code if:   * the request body is not in application/json. |

### Response Examples

#### Success

|  |
| --- |
| Status: 200 OK  {      "name": "Sea Witch - Modified",      "loads": [          {              "id": 5747559041597440,              "self": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/loads/5747559041597440"          }      ],      "type": "Catamaran",      "owner": "auth0|65746d84ef593280157a4ae3",      "length": 28,      "id": 5677209289228288,      "self": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/boats/5677209289228288"  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  {      "Error": "The request object is missing one of the required attributes."  }  Status: 400 Bad Request  {      "Error": "The request body contains unallowed attributes."  }  Status: 401 Unauthorized  {      "Error": "Missing or invalid JWTs."  }  Status: 403 Forbidden  {      "Error": "Valid JWT but the boat is owned by others."  }  Status: 404 Not Found  {      "Error": "No boat with this boat\_id exists."  }  Status: 406 Not Acceptable  {      "Error": "Request MIME type is not acceptable."  }  Status: 415 Unsupported Media Type  {      "Error": "Server only accepts application/json data."  } |

# Put a Boat (Required user JWT)

Allows you to edit all attributes of a particular boat. Cannot manually modify “loads” in this route.

|  |
| --- |
| PUT /boats/:boat\_id |

## Authorization

Bearer token

## 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. | Required |
| type | The type of the boat. E.g., Sailboat, Catamaran, etc. | Required |
| length | Length of the boat in feet. | Required |

### Request Body Example

|  |
| --- |
| {      "name": "Sea Witch - Modified by PUT",      "type": "Catamaran - Modified by PUT",      "length": 555  } |

## Response

### Response Body Format

JSON

### Response Statuses

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status Code** | **Notes** |
| Success | 200 OK | Return 200 status code if (pass all):   * only the allowed attribute(s) in the request body (except loads) * application/json for request body and return MIME type. * valid user JWT is provided. |
| Failure | 400 Bad Request | Return 400 status code if:   * invalid change of “loads” in this route. * Invalid or missing attribute(s) is found on the request body. |
| Failure | 401 Unauthorized | Return 401 status code if:   * no bearer token (user JWT) is provided. |
| Failure | 403 Forbidden | Return 403 status code if:   * the boat belongs to other user. |
| Failure | 404 Not Found | Return 404 status code if:   * invalid boat\_id is provided. |
| Failure | 406 Not Acceptable | Return 406 status code if:   * the client request response type is not in application/json. |
| Failure | 415 Unsupported Media Type | Return 415 status code if:   * the request body is not in application/json. |

### Response Examples

#### Success

|  |
| --- |
| Status: 200 OK  {      "type": "Catamaran - Modified by PUT",      "length": 555,      "loads": [],      "name": "Sea Witch - Modified by PUT",      "owner": "auth0|65746d84ef593280157a4ae3",      "id": 5646947083878400,      "self": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/boats/5646947083878400"  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  {      "Error": "The request object is missing one of the required attributes."  }  Status: 400 Bad Request  {      "Error": "The request body contains unallowed attributes."  }  Status: 400 Bad Request  {      "Error": "Cannot modify 'loads' in this route."  }  Status: 401 Unauthorized  {      "Error": "Missing or invalid JWTs."  }  Status: 403 Forbidden  {      "Error": "Valid JWT but the boat is owned by others."  }  Status: 404 Not Found  {      "Error": "No boat with this boat\_id exists."  }  Status: 406 Not Acceptable  {      "Error": "Request MIME type is not acceptable."  }  Status: 415 Unsupported Media Type  {      "Error": "Server only accepts application/json data."  } |

# 

# Delete a Boat (Required user JWT)

Allows you to delete a particular boat. Side effects (e.g., the carrier of the load will set to null if the boat is deleted) will also be done.

|  |
| --- |
| DELETE /boats/:boat\_id |

## Authorization

Bearer token

## 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 | Return 401 status code if:   * no bearer token (user JWT) is provided. |
| Failure | 403 Forbidden | Return 403 status code if:   * the boat belongs to other user. |
| Failure | 404 Not Found | Return 404 status code if:   * invalid boat\_id is provided. |

### Response Examples

#### Success

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

#### Failure

|  |
| --- |
| Status: 401 Unauthorized  {      "Error": "Missing or invalid JWTs."  }  Status: 403 Forbidden  {      "Error": "Valid JWT but the boat is owned by others."  }  Status: 404 Not Found  {      "Error": "No boat with this boat\_id exists."  } |

# Create a Load

Allows you to create a new load.

|  |
| --- |
| POST /loads |

## Request

### Path Parameters

None

### Request Body

Required

### Request Body Format

JSON

### Request JSON Attributes

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Required?** |
| volume | The volume of the load. | Required |
| item | The name / content of the load. | Required |
| creation\_date | The date that the load is created. | Required |

### Request Body Example

|  |
| --- |
| {    "volume": 5,    "item": "LEGO Blocks",    "creation\_date": "10/18/2021"  } |

## Response

### Response Body Format

JSON

### Response Statuses

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status Code** | **Notes** |
| Success | 201 Created |  |
| Failure | 400 Bad Request | Return 400 status code if:   * missing one of the required attributes. |
| Failure | 406 Not Acceptable | Return 406 status code if:   * the client request response type is not in application/json. |
| Failure | 415 Unsupported Media Type | Return 415 status code if:   * the request body is not in application/json. |

### Response Examples

* Datastore will automatically generate an ID and store it with the entity being created.
* The “carrier” attribute should be null in default. Adding boat into “carrier” can be performed in other route.
* The “self” attribute will not be stored in Datastore, but for displaying in the response body only.

#### Success

|  |
| --- |
| Status: 201 Created  {      "volume": 5,      "item": "LEGO Blocks",      "creation\_date": "10/18/2021",      "carrier": **null**,      "id": 5151970692169728,      "self": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/loads/5151970692169728"  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  {      "Error": "The request object is missing one of the required attributes."  }  Status: 406 Not Acceptable  {      "Error": "Request MIME type is not acceptable."  }  Status: 415 Unsupported Media Type  {      "Error": "Server only accepts application/json data."  } |

# Get all Loads

Allows you to get all 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

* A maximum of 5 records will be fetched per request (for pagination). The “next” attribute will direct to the following 5 loads.
* The “self” attribute will not be stored in Datastore, but for displaying in the response body only.

#### Success

|  |
| --- |
| Status: 200 OK  {      "loads": [          {              "carrier": **null**,              "item": "LEGO Blocks",              "volume": 5,              "creation\_date": "10/18/2021",              "id": 5151970692169728,              "self": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/loads/5151970692169728"          },          {              "volume": 3,              "carrier": **null**,              "item": "Food",              "creation\_date": "10/19/2021",              "id": 5657512099446784,              "self": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/loads/5657512099446784"          },          {              "volume": 4,              "carrier": **null**,              "item": "Toys",              "creation\_date": "10/21/2021",              "id": 5671985870798848,              "self": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/loads/5671985870798848"          },          {              "item": "Furnitures",              "volume": 7,              "creation\_date": "10/20/2021",              "carrier": **null**,              "id": 5675729136123904,              "self": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/loads/5675729136123904"          },          {              "item": "Football",              "volume": 6,              "creation\_date": "10/29/2023",              "carrier": **null**,              "id": 5686575606267904,              "self": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/loads/5686575606267904"          }      ],      "next": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/loads?limit=5&offset=5"  } |

# 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 | Return 404 status code if:   * invalid load\_id is provided. |
| Failure | 406 Not Acceptable | Return 406 status code if:   * the client request response type is not in application/json. |

### Response Examples

* The “self” attribute will not be stored in Datastore, but for displaying in the response body only.

#### Success

|  |
| --- |
| {      "carrier": **null**,      "creation\_date": "10/18/2021",      "volume": 5,      "item": "LEGO Blocks",      "id": 5151970692169728,      "self": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/loads/5151970692169728"  } |

#### Failure

|  |
| --- |
| Status: 404 Not Found  {      "Error": "No load with this load\_id exists."  }  Status: 406 Not Found  {      "Error": "Request MIME type is not acceptable."  } |

# Patch a Load

Allows you to edit 1 or more attribute(s) of a particular load. Cannot manually modify “carrier” in this route.

|  |
| --- |
| 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?** |
| volume | The volume of the load. | Optional |
| item | The name / content of the load. | Optional |
| creation\_date | The date that the load is created. | Optional |

### Request Body Example

|  |
| --- |
| {    "volume": 50,    "item": "Stuffed Animals - Modified by PATCH"  } |

## Response

### Response Body Format

JSON

### Response Statuses

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status Code** | **Notes** |
| Success | 200 OK | Return 200 status code if (pass all):   * only the allowed attribute(s) in the request body (except carrier) * application/json for request body and return MIME type. |
| Failure | 400 Bad Request | Return 400 status code if:   * invalid change of “carrier” in this route. * Invalid attribute(s) is found on the request body. |
| Failure | 404 Not Found | Return 404 status code if:   * invalid load\_id is provided. |
| Failure | 406 Not Acceptable | Return 406 status code if:   * the client request response type is not in application/json. |
| Failure | 415 Unsupported Media Type | Return 415 status code if:   * the request body is not in application/json. |

### Response Examples

#### Success

|  |
| --- |
| Status: 200 OK  {      "item": "Stuffed Animals - Modified by PATCH",      "carrier": **null**,      "volume": 100,      "creation\_date": "12/25/2023",      "id": 5151970692169728,      "self": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/loads/5151970692169728"  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  {      "Error": "The request body contains unallowed attributes."  }  Status: 400 Bad Request  {      "Error": "Cannot modify 'carrier' in this route."  }  Status: 403 Forbidden  {      "Error": "Valid JWT but the boat is owned by others."  }  Status: 404 Not Found  {      "Error": "No load with this load\_id exists."  }  Status: 406 Not Acceptable  {      "Error": "Request MIME type is not acceptable."  }  Status: 415 Unsupported Media Type  {      "Error": "Server only accepts application/json data."  } |

# Put a Load

Allows you to edit all attributes of a particular load. Cannot manually modify “carrier” in this route.

|  |
| --- |
| 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?** |
| volume | The volume of the load. | Required |
| item | The name / content of the load. | Required |
| creation\_date | The date that the load is created. | Required |

### Request Body Example

|  |
| --- |
| {    "volume": 999,    "item": "Stuffed Animals - Modified by PUT",    "creation\_date": "12/1/2023"  } |

## Response

### Response Body Format

JSON

### Response Statuses

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status Code** | **Notes** |
| Success | 200 OK | Return 200 status code if (pass all):   * only the allowed attribute(s) in the request body (except carrier) * application/json for request body and return MIME type. |
| Failure | 400 Bad Request | Return 400 status code if:   * invalid change of “carrier” in this route. * Invalid / missing attribute(s) is found on the request body. |
| Failure | 404 Not Found | Return 404 status code if:   * invalid load\_id is provided. |
| Failure | 406 Not Acceptable | Return 406 status code if:   * the client request response type is not in application/json. |
| Failure | 415 Unsupported Media Type | Return 415 status code if:   * the request body is not in application/json. |

### Response Examples

#### Success

|  |
| --- |
| Status: 200 OK  {      "carrier": **null**,      "creation\_date": "12/1/2023",      "volume": 999,      "item": "Stuffed Animals - Modified by PUT",      "id": 5151970692169728,      "self": "https://cs493-protfolio-lottotlt.uc.r.appspot.com/loads/5151970692169728"  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  {      "Error": "The request object is missing one of the required attributes."  }  Status: 400 Bad Request  {      "Error": "The request body contains unallowed attributes."  }  Status: 400 Bad Request  {      "Error": "Cannot modify 'carrier' in this route."  }  Status: 404 Not Found  {      "Error": "No load with this load\_id exists."  }  Status: 406 Not Acceptable  {      "Error": "Request MIME type is not acceptable."  }  Status: 415 Unsupported Media Type  {      "Error": "Server only accepts application/json data."  } |

# Delete a Load

Allows you to delete a particular load. Side effects (e.g., the loads of the boat will also be deleted if the load is deleted) will also be done.

|  |
| --- |
| 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 | Return 404 status code if:   * invalid load\_id is provided. |

### Response Examples

#### Success

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

#### Failure

|  |
| --- |
| Status: 404 Not Found  {      "Error": "No load with this load\_id exists."  } |

# Managing Loads - Assign Load to a Boat (Required user JWT)

Assign load to a boat. Only the boat owner can perform this operation.

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

## Authorization

Bearer token

## 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 | 401 Unauthorized | Return 401 status code if:   * no bearer token (user JWT) is provided. |
| Failure | 403 Forbidden | Return 403 status code if:   * the boat belongs to other user. * the load is already loaded on other boat. |
| Failure | 404 Not Found | Return 404 status code if:   * invalid boat\_id is provided. * Invalid load\_id is provided. |

### Response Examples

#### Success

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

#### Failure

|  |
| --- |
| Status: 401 Unauthorized  {      "Error": "Missing or invalid JWTs."  }  Status: 403 Forbidden  {      "Error": "Valid JWT but the boat is owned by others."  }  Status: 403 Forbidden  {      "Error": "The load is already loaded on another boat."  }  Status: 404 Not Found  {      "Error": "No load with this load\_id exists."  }  Status: 404 Not Found  {      "Error": "No boat with this boat\_id exists."  } |

## Comment

* A load can only be assigned in a boat; a boat can have multiple loads.

# Managing Loads - Remove Load from a Boat (Required user JWT)

Unload load from the boat. Only the boat owner can perform this operation.

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

## Authorization

Bearer token

## 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 | 401 Unauthorized | Return 401 status code if:   * no bearer token (user JWT) is provided. |
| Failure | 403 Forbidden | Return 403 status code if:   * the boat belongs to other user. |
| Failure | 404 Not Found | Return 404 status code if:   * invalid boat\_id is provided. * Invalid load\_id is provided. * valid boat and load, but cannot find the load in that boat. |

### Response Examples

#### Success

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

#### Failure

|  |
| --- |
| Status: 401 Unauthorized  {      "Error": "Missing or invalid JWTs."  }  Status: 403 Forbidden  {      "Error": "Valid JWT but the boat is owned by others."  }  Status: 404 Not Found  {      "Error": "No load with this load\_id is assigned on the boat on this boat\_id."  }  Status: 404 Not Found  {      "Error": "No load with this load\_id exists."  }  Status: 404 Not Found  {      "Error": "No boat with this boat\_id exists."  } |

# Miscellaneous – Unallowed HTTP verb for routes not stated above

Error message will be prompted for using the unallowed HTTP verb on the routes that are not registered on the App.

## Request

### Path Parameters

None

### Request Body

None

## Response

No body

### Response Body Format

Failure: JSON

### Response Statuses

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status Code** | **Notes** |
| Failure | 405 Method Not Allowed | Return 405 status code if:   * using the unallowed HTTP verb for the route. |

### Response Examples

#### Failure

|  |
| --- |
| Status: 405 Method Not Allowed  {      "Error": "Method not recognized."  } |