

Final Project: API Spec

CS 493: Cloud Application Development

Fall 2021

Oregon State University

Last Update: Dec 3, 2021. 7:16 pm Pacific

Change log	1
Application URL.....	2
User Login URL.....	2
Data Model	2
Get all Users (Unprotected)	4
Create a Boat (Protected)	5
Get a Boat (Protected)	7
List all Boats (Protected)	9
Edit a Boat with PUT (Protected)	12
Edit a Boat with PATCH (Protected)	14
Delete a Boat (Protected)	16
Create a Load (Unprotected)	17
Get a Load (Unprotected)	19
Get all Loads (Unprotected)	21
Update a Load with PUT (Unprotected)	23
Update a Load with PATCH (Unprotected)	25
Delete a Load (Unprotected)	27
Assign a Load to a Boat (Unprotected)	28
Remove a Load from a Boat (Unprotected)	29
Unsupported Methods	30

Change log

Version	Change	Date
1.0	Initial version.	Dec 3, 2021

Application URL

app_url: <https://liumiche-portfolio.wl.r.appspot.com>

User Login URL

url: <https://liumiche-portfolio.wl.r.appspot.com>

Data Model

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

Boats

- The relationship between Boats and Loads is one-to-many. A boat can carry multiple loads at a time, but a load can only be loaded on one boat at a time.
- Every boat must have an owner upon creation. The value of owner is the “sub” property of the provided JWT.
- Boat names don’t have to be unique.

Property	Data Type	Notes
id	Integer	The id of the boat. Automatically generated by Datastore.
name	String	Name of the boat. Only contains characters in [a-z, A-Z, ' '] and contains at most 30 characters.
type	String	Type of the boat. E.g., Sailboat, Catamaran, etc. Only contains characters in [a-z, A-Z, ' '], and contains at most 30 characters.
length	Integer	The length of the boat in feet.
loads	Array	If the boat has any loads, then integer array of the ids of those loads. Otherwise, [].
owner	String	The value of user_id of a user, or the “sub” property of a supplied JWT.

Loads

Property	Data Type	Notes
id	Integer	The id of the load. Automatically generated by Datastore.
volume	Integer	The volume of the load.
content	String	The Content of the load. Only contains characters in [a-z, A-Z, ' '], and contains at most 30 characters.
creation_date	String	The date the load was created in the format MM/DD/YYYY.
carrier	Integer	If the load is on a boat, then id of that boat. Otherwise, null.

Users

- The unique identifier of a user is the “sub” property of a JWT.
- The relationship between Users and Boats is one-to-many. Every boat is owned by a user in the application, but a user can own multiple boats.
- If a request needs to supply a user identifier, the person making the request needs to send a valid JWT for the authenticated user along with the request (Select bearer token under the Authorization tab in Postman, and supply the JWT value in the token field).
- If a request is supplied with a valid JWT, the “sub” property of the JWT will be used to map the JWT to a user using the “user_id” property of the user.

Property	Data Type	Notes
id	Integer	The id of the user. Automatically generated by Datastore.
first_name	String	The first name of the user, or the “given_name” property of a JWT.
last_name	String	The last name of the user, or the “family_name” property of a JWT.
user_id	String	The unique user ID of the user, or the “sub” property of a JWT.

Get all Users (Unprotected)

List all 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

- [] will be returned if there is no users in the application.

Success

Status: 200 OK

```
[
{
  "id": 123,
  "first_name": "John",
  "last_name": "Doe",
  "user_id": "123456789"
},
{
  "id": 456,
  "first_name": "Emily",
  "last_name": "Doe",
  "user_id": "123456789"
},
{
  "id": 789,
  "first_name": "James",
  "last_name": "Bond",
  "user_id": "987654321"
}
]
```

Create a Boat (Protected)

Allows you to create a new boat that belongs to the user authenticated by the JWT 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>You don't need to validate the values of the attributes and can assume that if the request contains any of the listed attributes, then that attribute's 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 that is not listed above).</p>
Failure	401 Unauthorized	401 status code must be returned if the request is missing a valid JWT or contains an invalid JWT.

Failure	406 Not Acceptable	406 status code must be returned if the accept header of the request doesn't include JSON.
---------	--------------------	--

Response Examples

- An ID is generated by Datastore and is stored with the entity being created. The app must send back this value in the response body as shown in the example.
- The `self` attribute will contain the live link to the REST resource corresponding to this boat. In other words, this is the URL to get this newly created boat. Note that the `self` attribute is not stored in Datastore

Success

Status: 201 Created

```
{
  "id": 123,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "loads": [],
  "owner": "123456789",
  "self": "https://<app-url>/boats/123"
}
```

Failure

Status: 400 Bad Request

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

Status: 401 Unauthorized

```
{
  "Error": "The request is missing a valid JWT or contains an invalid JWT"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Verify accept header. The server only offers response in JSON"
}
```

Get a Boat (Protected)

Allows you to get an existing boat that belongs to the user authenticated by the JWT provided.

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	401 status code must be returned if the request is missing a valid JWT or contains an invalid JWT.
Failure	403 Forbidden	Return 403 status code <ul style="list-style-type: none">If the JWT is valid but boat_id is owned by someone elseIf the JWT is valid but no boat with this boat_id exists
Failure	406 Not Acceptable	406 status code must be returned if the accept header of the request doesn't include JSON.

Response Examples

Success

Status: 200 OK

```
{
  "id": 123,
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "owner": "123456789",
  "loads": [
    {
      "id": 4567,
      "self": "https://<app-url>/loads/4567"
    },
    {
      "id": 7654,
      "self": "https://<app-url>/loads/7654"
    }
  ]
}
```

```
],  
  "self": "https://<app-url>/boats/123"  
}
```

Failure

Status: 401 Unauthorized

```
{  
  "Error": "The request is missing a valid JWT or contains an invalid JWT"  
}
```

Status: 403 Forbidden

```
{  
  "Error": "The boat is owned by someone else or no boat with this boat_id exists"  
}
```

Status: 406 Not Acceptable

```
{  
  "Error": "Verify accept header. The server only offers response in JSON"  
}
```


List all Boats (Protected)

List all the boats that belong to the user authenticated by the JWT provided.

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	401 status code must be returned if the request is missing a valid JWT or contains an invalid JWT.
Failure	406 Not Acceptable	406 status code must be returned if the accept header of the request doesn't include JSON.

Response Examples

- The response should implement pagination and return 5 loads per page.
- There must be a 'next' link to get the next page of results, unless there are no more pages of result (i.e., the last page of results shouldn't have the 'next' link).
- If there is no boat, return an empty list for the attribute `boats`.
- The value of the `total_items` attribute represents the number of boats owned by the user authenticated by the provided JWT.

Success

Status: 200 OK

```
{
  "boats": [
    {
      "id": 123,
      "name": "Sea Witch",
      "type": "Catamaran",
      "length": 28,
      "owner": "123456789",
      "loads": [
        {
          "id": 1234,
```

```

    "self": " https://<app-url>/loads/1234"
  }
],
  "self": "https://<app-url>/boats/123"
},
{
  "id": 456,
  "name": "Sea Witch II",
  "type": "Catamaran",
  "length": 32,
  "owner": "123456789",
  "loads": [],
  "self": "https://<app-url>/boats/456"
},
{
  "id": 789,
  "name": "Sea Witch III",
  "type": "Catamaran",
  "length": 39,
  "owner": "123456789",
  "loads": [],
  "self": "https://<app-url>/boats/789"
},
{
  "id": 987,
  "name": "Sea Witch IV",
  "type": "Catamaran",
  "length": 30,
  "owner": "123456789",
  "loads": [],
  "self": "https://<app-url>/boats/987"
},
{
  "id": 765,
  "name": "Sea Witch V",
  "type": "Catamaran",
  "length": 27,
  "owner": "123456789",
  "loads": [],
  "self": "https://<app-url>/boats/765"
}
],
"total_items": 6,
"next": "https://<app-url>/boats?limit=5&offset=5"
}

```

Failure

Status: 401 Unauthorized

```
{  
"Error": "The request is missing a valid JWT or contains an invalid JWT"  
}
```

Status: 406 Not Acceptable

```
{  
"Error": "Verify accept header. The server only offers response in JSON"  
}
```

Edit a Boat with PUT (Protected)

Allows you to edit a boat that belongs to the user authenticated by the JWT 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 II",
  "type": "Catamaran",
  "length": 99
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	If the request is missing any of the 3 required attributes, the boat must not be updated, and 400 status code must be returned. You don't need to validate the values of the attributes and can assume that if the request contains any of the listed attributes, then that attribute's value is valid. 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 the ones that are listed).
Failure	401 Unauthorized	401 status code must be returned if the request is missing a valid JWT or contains an invalid JWT.
Failure	403 Forbidden	Return 403 status code:

		<ul style="list-style-type: none"> • If the JWT is valid but boat_id is owned by someone else • If the JWT is valid but no boat with this boat_id exists
Failure	406 Not Acceptable	406 status code must be returned if the accept header of the request doesn't include JSON.

Response Examples

- The `self` attribute will contain the live link to the REST resource corresponding to this boat. Note that `self` is not stored in Datastore.
- To see loads on a boat, please go to the boat's `self` link for more detail.

Success

Status: 200 OK

```
{
  "id": 123,
  "name": "Sea Witch II",
  "type": "Catamaran",
  "length": 99,
  "owner": "123456789",
  "self": "https://<app-url>/boats/123"
}
```

Failure

Status: 400 Bad Request

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

Status: 401 Unauthorized

```
{
  "Error": "The request is missing a valid JWT or contains an invalid JWT"
}
```

Status: 403 Forbidden

```
{
  "Error": "The boat is owned by someone else or no boat with this boat_id exists"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Verify accept header. The server only offers response in JSON"
}
```

Edit a Boat with PATCH (Protected)

Allows you to edit a boat that belongs to the user authenticated by the JWT 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

```
{
  "name": "Sea Witch III",
  "type": "Yacht"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	<p>If the request body does not contain any of the 3 valid attributes, the boat must not be updated, and 400 status code must be returned.</p> <p>You don't need to validate the values of the attributes and can assume that if the request contains any of the listed attributes, then that attribute's 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 the ones that are listed).</p>
Failure	401 Unauthorized	401 status code must be returned if the request is missing a valid JWT or contains an invalid JWT.
Failure	403 Forbidden	Return 403 status code:

		<ul style="list-style-type: none"> • If the JWT is valid but boat_id is owned by someone else • If the JWT is valid but no boat with this boat_id exists
Failure	406 Not Acceptable	406 status code must be returned if the accept header of the request doesn't include JSON.

Response Examples

- The `self` attribute will contain the live link to the REST resource corresponding to this boat. Note that `self` is not stored in Datastore.
- To see loads on a boat, please go to the boat's `self` link for more detail.

Success

Status: 200 OK

```
{
  "id": 123,
  "name": "Sea Witch III",
  "type": "Yacht",
  "length": 99,
  "owner": "123456789",
  "self": "https://<app-url>/boats/123"
}
```

Failure

Status: 400 Bad Request

```
{
  "Error": "The request object does not contain any of the valid attributes"
}
```

Status: 401 Unauthorized

```
{
  "Error": "The request is missing a valid JWT or contains an invalid JWT"
}
```

Status: 403 Forbidden

```
{
  "Error": "The boat is owned by someone else or no boat with this boat_id exists"
}
```

Status: 406 Not Acceptable

```
{
  "Error": "Verify accept header. The server only offers response in JSON"
}
```

Delete a Boat (Protected)

Allows you to delete a boat. Note that deleting a boat will unload any loads that's on the boat.

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	401 status code must be returned if the request is missing a valid JWT or contains an invalid JWT.
Failure	403 Forbidden	Return 403 status code: <ul style="list-style-type: none">If the JWT is valid but boat_id is owned by someone elseIf the JWT is valid but no boat with this boat_id exists

Response Examples

Success

Status: 204 No Content

Failure

Status: 401 Unauthorized

```
{
  "Error": "The request is missing a valid JWT or contains an invalid JWT"
}
```

Status: 403 Forbidden

```
{
  "Error": "The boat is owned by someone else or no boat with this boat_id exists"
}
```


Create a Load (Unprotected)

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.	Yes
content	The content of the load.	Yes
creation_date	The date the load was created in the format MM/DD/YYYY.	Yes

Request Body Example

```
{
  "volume": 5,
  "content": "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	<p>If the request is missing any of the three 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 the attributes and can assume that if the request contains any of the listed attributes, then that attribute's 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 attributes other than the ones specified).</p>
Failure	406 Not Acceptable	406 status code must be returned if the accept header of the request doesn't include JSON.

Response Examples

- Datastore will automatically generate an ID and store it with the entity being created. This value needs to be sent in the response body as shown in the example.
- The value of the attribute `carrier` is the ID of the boat currently carrying the load. If there is no boat carrying this load, the value of `carrier` should be null.
- The value of the attribute `self` is a live link to the REST resource corresponding to this load. Note that `self` is not stored in Datastore.

Success

Status: 201 Created

```
{
  "id": 5678,
  "volume": 5,
  "carrier": null,
  "content": "LEGO Blocks",
  "creation_date": "10/18/2020",
  "self": "https://<app-url>/loads/5678"
}
```

Failure

Status: 400 Bad Request

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

Status: 406 Not Acceptable

```
{
  "Error": "Verify accept header. The server only offers response in JSON"
}
```

Get a Load (Unprotected)

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	406 Not Acceptable	406 status code must be returned if the accept header of the request doesn't include JSON.

Response Examples

- If a load is not assigned to a boat, the carrier field will display null.
- The `self` attribute will contain the live link to the REST resource corresponding to this load.
Note that `self` is not stored in Datastore.

Success

Status: 200 OK

```
{
  "id": 1234,
  "volume": 5,
  "carrier": {
    "id": 123,
    "name": "Sea Witch",
    "self": " https://<app-url>/boats/123"
  },
  "content": "LEGO Blocks",
  "creation_date": "10/18/2020",
  "self": "https://<app-url>/loads/1234"
}
```

Failure

Status: 404 Not Found

```
{
```

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

Status: 406 Not Acceptable

```
{
  "Error": "Verify accept header. The server only offers response in JSON"
}
```

Get all Loads (Unprotected)

Allows you to get all existing loads.

GET /loads

Request

Path Parameters

None

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	406 Not Acceptable	406 status code must be returned if the accept header of the request doesn't include JSON.

Response Examples

- The response should implement pagination and return 5 loads per page.
- There must be a 'next' link to get the next page of results, unless there are no more pages of result (i.e., the last page of results shouldn't have the 'next' link).
- If a load is not assigned to a boat, its carrier field will display null.
- If there is no load, return an empty list for the attribute loads.

Success

Status: 200 OK

```
{
  "loads": [
    {
      "id": 1234,
      "volume": 5,
      "carrier": {
        "id": 123,
        "self": "https://<app-url>/boats/123"
      },
      "content": "LEGO Blocks",
      "creation_date": "10/18/2020",
      "self": "https://<app-url>/loads/1234"
    },
    {
      "id": 2345,
      "volume": 3,
      "carrier": {
```

```
    "id": 123,
    "self": "https://<app-url>/boats/123"
  },
  "content": "Toys",
  "creation_date": "10/18/2020",
  "self": "https://<app-url>/loads/2345"
},
{
  "id": 3456,
  "volume": 2,
  "carrier": {
    "id": 123,
    "self": "https://<app-url>/boats/123"
  },
  "content": "Puzzles",
  "creation_date": "10/18/2020",
  "self": "https://<app-url>/loads/3456"
},
{
  "id": 4567,
  "volume": 6,
  "carrier": {
    "id": 123,
    "self": "https://<app-url>/boats/123"
  },
  "content": "Clothing",
  "creation_date": "10/18/2020",
  "self": "https://<app-url>/loads/4567"
},
{
  "id": 5678,
  "volume": 5,
  "carrier": {
    "id": 123,
    "self": "https://<app-url>/boats/123"
  },
  "content": "Books",
  "creation_date": "10/18/2020",
  "self": "https://<app-url>/loads/5678"
}
],
"total_items": 6,
"next": "https://<app-url>/boats?limit=5&offset=5"
}
```

Failure

Status: 406 Not Acceptable

```
{
  "Error": "Verify accept header. The server only offers response in JSON"
}
```

Update a Load with PUT (Unprotected)

Allows you to edit a 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?
volume	The volume of the load.	Yes
content	The content of the load.	Yes
creation_date	The date the load was created in the format MM/DD/YYYY.	Yes

Request Body Example

```
{
  "volume": 3,
  "content": "LEGO Blocks",
  "creation_date": "10/30/2021"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	<p>If the request is missing any of the 3 required attributes, the load must not be updated, and 400 status code must be returned.</p> <p>You don't need to validate the values of the attributes and can assume that if the request contains any of the listed attributes, then that attribute's 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 the ones that are listed).</p>
Failure	404 Not Found	No load with this load_id exists

Failure	406 Not Acceptable	406 status code must be returned if the accept header of the request doesn't include JSON.
---------	--------------------	--

Response Examples

- The `self` attribute will contain the live link to the REST resource corresponding to this load.
Note that `self` is not stored in Datastore.
- If a load is not assigned to a boat, its carrier field will display null.

Success

Status: 200 OK

```
{
  "id": 1234,
  "volume": 3,
  "carrier": {
    "id": 123,
    "self": " https://<app-url>/boats/123"
  },
  "content": "LEGO Blocks",
  "creation_date": "10/30/2020",
  "self": "https://<app-url>/loads/1234"
}
```

Failure

Status: 400 Bad Request

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

Status: 404 Not Found

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

Status: 406 Not Acceptable

```
{
  "Error": "Verify accept header. The server only offers response in JSON"
}
```


Update a Load with PATCH (Unprotected)

Allows you to edit a load.

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.	No
content	The content of the load.	No
creation_date	The date the load was created in the format MM/DD/YYYY.	No

Request Body Example

```
{
  "volume": 7,
  "creation_date": "10/20/2021"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	<p>If the request body does not contain any of the 3 valid attributes, the load must not be updated and 400 status code must be returned.</p> <p>You don't need to validate the values of the attributes and can assume that if the request contains any of the listed attributes, then that attribute's 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 the ones that are listed).</p>
Failure	404 Not Found	No load with this load_id exists

Failure	406 Not Acceptable	406 status code must be returned if the accept header of the request doesn't include JSON.
---------	--------------------	--

Response Examples

- The `self` attribute will contain the live link to the REST resource corresponding to this load.
Note that `self` is not stored in Datastore.
- If a load is not assigned to a boat, its carrier field will display null.

Success

Status: 200 OK

```
{
  "id": 1234,
  "volume": 7,
  "carrier": {
    "id": 123,
    "self": " https://<app-url>/boats/123"
  },
  "content": "LEGO Blocks",
  "creation_date": "10/20/2020",
  "self": "https://<app-url>/loads/1234"
}
```

Failure

Status: 400 Bad Request

```
{
  "Error": "The request object does not contain any of the valid attributes"
}
```

Status: 404 Not Found

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

Status: 406 Not Acceptable

```
{
  "Error": "Verify accept header. The server only offers response in JSON"
}
```

Delete a Load (Unprotected)

Allows you to delete a load. If the load being deleted is on a boat, the load will be unloaded from the boat.

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"
}
```

Assign a Load to a Boat (Unprotected)

A load is loaded onto a boat.

PUT /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

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 currently loaded on any boat.
Failure	403 Forbidden	The load is already loaded on another boat.
Failure	404 Not Found	The specified boat and/or load does not exist

Response Examples

- A load cannot be assigned to multiple slips.

Success

Status: 204 No Content

Failure

Status: 403 Forbidden

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

Status: 404 Not Found

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

Remove a Load from a Boat (Unprotected)

Unload 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	Succeeds only if a boat exists with this boat_id, a load exists with this load_id and the load is loaded on the boat.
Failure	404 Not Found	No boat with this boat_id is loaded with the load with this load_id.

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found

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

Unsupported Methods

Any method not included in the spec is not supported, e.g. DELETE /boats, PUT /boats, GET /boats/:boat_id/loads/:load_id, etc.

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Failure	405 Method Not Allowed	405 Method Not Allowed.

Response Examples

Failure

Status: 405 Method Not Allowed
<pre>{ "Error": "405 Method Not Allowed" }</pre>