Michael Chen
CS493 Portfolio Assignment
12/9/2023

Table of Contents

API URL: https://clean-utility-402919.uw.r.appspot.com

Login URL: https://clean-utility-402919.uw.r.appspot.com

Data Model

The app stores Users, Boats, and Slips entities in Datastore.

Users

Property	Data type	Description
id	integer	Required. Integer id number automatically generated by Datastore.
user_sub	string	Required. Unique string identifier generated from Auth0 jwt. Valid values are generated by Auth0 and able to be read from validated jwt.

The unique identifier for the User is the user_sub property from the jwt generated by Auth0. If a request needs a user identifier, it must have a jwt with the correct user_sub. Provided jwt is mapped to the user_sub identifier by verifying the jwt and extracting the user_sub property from the provided jwt.

Users are populated upon arrival at the callback URL after logging into Auth0. Note for creating Boats, it is necessary for the user to have been logged in and created via the website callback after logging in.

User – Boat Relationship

Users have an implicit one to many relationship with Boats, which is defined by Auth0 jwts instead of an explicit Datastore property. I acknowledge this is probably not a realistic design, but I do not see where an explicit ‘boats owned by user’ property is required in Users by the assignment. The assignment specification did not require any operations on Users which may affect Boats.

Boats must explicitly have a User defined in a user_sub property, and Users optionally have Boats. All Boat operations must have jwt authorization, which identifies a User by user_sub property. Therefore I found it unnecessary to have an explicit property in Users to show Boats associated with a User, while it is necessary to have an explicit property in Boats to associate a User.

Boats

Property	**Data type**	**Description**
id	integer	Required. Integer id number automatically generated by Datastore.
owner	string	Required. Unique string identifier (user_sub) of a User generated from Auth0 jwt. Valid values are generated by Auth0 and able to be read from validated jwt.
name	string	Required. Valid input is a string representing the name of the Boat.
type	string	Required. Valid input is a string representing the type of the Boat.
length	integer	Required. Valid input is an integer representing the length of a Boat.
slip	object	Required. Either None if Boat is not at a Slip, or an object with {“id”: <int>slip_id}.

Slips

Property	**Data type**	**Description**
id	integer	Integer id number automatically generated by Datastore.
harbor	string	Required. Valid input is a string representing the name of a harbor hosting the Slip.
depth	integer	Required. Valid input is an integer representing the mean low low water depth of the Slip.
rate	integer	Required. Valid input is an integer representing the fee charge rate to dock a Boat at this Slip.
boat	object	Required. Either None if no Boat is not at this Slip, or an object with {“id”: <int>boat_id}.

Boat – Slip Relationship

Boats and Slips have an optional one to one relationship. One singular boat may be docked at one singular slip. This is explicitly reflected in the properties for both Boats and Slips.

Get Users

Gets all the users that have logged into the app.

GET /users

Request

Path Parameters

None

Request Body

None

Request Body Format

N/A

Request JSON Attributes

N/A

Request Body Example

N/A

Response

Response Body Format

JSON *

Response Statuses

Outcome	**Status Code**	**Notes**
Success	200 OK	
Failure	405 Method Not Allowed	Only GET requests are allowed. Automatic response by python Flask, in text/html format. (*)
Failure	406 Not Acceptable	Request did not have accept application/json header.

*** I could correct the 405 error format to application/json in Flask by tricking Flask into accepting the not-acceptable methods in the decorator keyword, but then returning the 405 Method Not Allowed with application/json in my API code. However I feel this would be silly. Another way would be to fork Flask and custom change its default response, but I did not feel this is in the scope of the course. This applies to all endpoints response with status code 405.**

Response Examples

Success. Status: 200 OK

```
[
{
"user_sub": "auth0|655c254031108fd1ba1599af",
"id": 5675729136123904
}
]
```

Failure. Status: 405 Method Not Allowed

```
<!doctype html>
<html lang=en>
<title>405 Method Not Allowed</title>
<h1>Method Not Allowed</h1>
<p>The method is not allowed for the requested URL.</p>
```

Failure. Status 406 Not Acceptable

```
{
"Error": "application/json response only"
}
```

Post Boats

Create a boat with a jwt.

POST /boats

Request

Path Parameters

None

Request Headers (required)

```
Accept: application/json

Authorization: <jwt>
```

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Required	Description
name	Yes	<string> The name of the boat
type	Yes	<string> The type of the boat.
length	Yes	<integer> The length of the boat.

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	Request body was missing required attributes.
Failure	401 Unauthorized	Request header was missing authorization jwt.
Failure	403 Forbidden	I return a 403 in the scenario where a jwt validates, but the user was not registered in Datastore through the login process. **That is to say you must get your jwt's from completing the login process.** This one is a little bit weird and I couldn't figure out a realistic way to test it. Please see another 403 for the postman test example.
Failure	405 Method Not Allowed	Request method not allowed by endpoint. Automatic response by python Flask, in text/html format.
Failure	406 Not Acceptable	Request did not have accept application/json header.

Response Examples

Success. Status: 201 Created

```
{
"name": "Sea Witch",
"type": "Catamaran",
"length": 28,
"slip": null,
"owner": "auth0|655c254031108fd1ba1599af",
"id": 5190718477828096,
"self": "https://clean-utility-402919.uw.r.appspot.com/boats/5190718477828096"
}
```

Failure. Status: 400 Bad Request

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

Failure. Status: 401 Unauthorized

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

Failure. Status: 403 Forbidden

```
{
"Error": "User not registered"
}
```

Failure. Status: 405 Method Not Allowed

```
<!doctype html>
<html lang=en>
<title>405 Method Not Allowed</title>
<h1>Method Not Allowed</h1>
<p>The method is not allowed for the requested URL.</p>
```

Failure. Status 406 Not Acceptable

```
{
"Error": "application/json response only"
}
```

Get Boats

Get all the boats associated with a user (validated by jwt). In the case where the jwt is validated by the user is not associated with any boats, then no boats are returned.

GET /boats?offset=<offset>&limit=<limit>

Request

Path Parameters

Name	Required	Description
offset	No	<string> String representation of an <integer> for pagination offset. Default value = 0.
limit	No	<string> String representation of an <integer> for pagination limit. Default value = 5.

Request Headers (required)

```
Accept: application/json

Authorization: <jwt>
```

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	Request header was missing authorization jwt.
Failure	405 Method Not Allowed	Request method not allowed by endpoint. Automatic response by python Flask, in text/html format.
Failure	406 Not Acceptable	Request did not have accept application/json header.

Response Examples

Success. Status: 200 OK

```
{
"boats": [
{
"slip": null,
"name": "Sea Witch",
"type": "Catamaran",
"length": 28,
"owner": "auth0|655c254031108fd1ba1599af",
"id": 5190718477828096,
"self": "https://clean-utility-402919.uw.r.appspot.com/boats/5190718477828096"
}
],
"next": null
}
```

Failure. Status: 401 Unauthorized

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

Failure. Status: 405 Method Not Allowed

```
<!doctype html>
<html lang=en>
<title>405 Method Not Allowed</title>
<h1>Method Not Allowed</h1>
<p>The method is not allowed for the requested URL.</p>
```

Failure. Status 406 Not Acceptable

```
{
"Error": "application/json response only"
}
```

Get Boat

Get a particular boat. Must be authorized by the boat user’s jwt.

GET /boats/<boat_id>

Request

Path Parameters (optional)

Name	Required	Description
boat_id	Yes	<string> String representation of an <integer> id of a boat.

Request Headers (required)

```
Accept: application/json

Authorization: <jwt>
```

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	Request header was missing authorization jwt.
Failure	403 Forbidden	Request jwt was validated, but not the correct user associated with the requested boat. Boat is owned by someone else.
Failure	404 Not Found	Boat does not exist.
Failure	405 Method Not Allowed	Request method not allowed by endpoint. Automatic response by python Flask, in text/html format.
Failure	406 Not Acceptable	Request did not have accept application/json header.

Response Examples

Success. Status: 200 OK

```
{
"slip": null,
"name": "Sea Witch",
"type": "Catamaran",
"length": 28,
"owner": "auth0|655c254031108fd1ba1599af",
"id": 5190718477828096,
"self": "https://clean-utility-402919.uw.r.appspot.com/boats/5190718477828096"
}
```

Failure. Status: 401 Unauthorized

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

Failure. Status: 403 Forbidden

```
{
"Error": "Boat is owned by someone else."
}
```

Failure. Status: 404 Not Found

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

Failure. Status: 405 Method Not Allowed

```
<!doctype html>
<html lang=en>
<title>405 Method Not Allowed</title>
<h1>Method Not Allowed</h1>
<p>The method is not allowed for the requested URL.</p>
```

Failure. Status 406 Not Acceptable

```
{
"Error": "application/json response only"
}
```

Delete Boat

Delete a particular boat. Must be authorized by the boat user’s jwt.

DELETE /boats/<boat_id>

Request

Path Parameters (optional)

Name	**Required**	**Description**
boat_id	Yes	<string> String representation of an <integer> id of a boat.

Request Headers (required)

```
Authorization: <jwt>
```

Request Body

None

Response

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	**Status Code**	**Notes**
Success	204 No Content	
Failure	401 Unauthorized	Request header was missing authorization jwt.
Failure	403 Forbidden	Request jwt was validated, but not the correct user associated with the requested boat. Boat is owned by someone else.
Failure	404 Not Found	Boat does not exist.
Failure	405 Method Not Allowed	Request method not allowed by endpoint. Automatic response by python Flask, in text/html format.

Response Examples

Success. Status: 204 No Content

Failure. Status: 401 Unauthorized

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

Failure. Status: 403 Forbidden

```
{
"Error": "Boat is owned by someone else."
}
```

Failure. Status: 404 Not Found

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

Failure. Status: 405 Method Not Allowed

```
<!doctype html>
<html lang=en>
<title>405 Method Not Allowed</title>
<h1>Method Not Allowed</h1>
<p>The method is not allowed for the requested URL.</p>
```

Put Boat

Update all attributes of a particular boat. Must be authorized by the boat user’s jwt.

PUT /boats/<boat_id>

Request

Path Parameters (optional)

Name	**Required**	**Description**
boat_id	Yes	<string> String representation of an <integer> id of a boat.

Request Headers (required)

```
Accept: application/json

Authorization: <jwt>
```

Request Body

Required

Request Body Format

JSON

Request JSON Attribute

Name	**Required**	**Description**
name	Yes	<string> The name of the boat
type	Yes	<string> The type of the boat.
length	Yes	<integer> The length of the boat.

Response

Response Body Format

JSON

Response Statuses

Outcome	**Status Code**	**Notes**
Success	200 OK	
Failure	400 Bad Request	Request body was missing required attributes.
Failure	401 Unauthorized	Request header was missing authorization jwt.
Failure	403 Forbidden	Request jwt was validated, but not the correct user associated with the requested boat. Boat is owned by someone else.
Failure	404 Not Found	Boat does not exist.
Failure	405 Method Not Allowed	Request method not allowed by endpoint. Automatic response by python Flask, in text/html format.
Failure	406 Not Acceptable	Request did not have accept application/json header.

Response Examples

Success. Status: 200 OK

```
{
"name": "Sea Witch",
"type": "Catamaran",
"length": 28,
"slip": null,
"owner": "auth0|655c254031108fd1ba1599af",
"id": 5190718477828096,
"self": "https://clean-utility-402919.uw.r.appspot.com/boats/5190718477828096"
}
```

Failure. Status: 400 Bad Request

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

Failure. Status: 401 Unauthorized

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

Failure. Status: 403 Forbidden

```
{
"Error": "Boat is owned by someone else."
}
```

Failure. Status: 404 Not Found

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

Failure. Status: 405 Method Not Allowed

```
<!doctype html>
<html lang=en>
<title>405 Method Not Allowed</title>
<h1>Method Not Allowed</h1>
<p>The method is not allowed for the requested URL.</p>
```

Failure. Status 406 Not Acceptable

```
{
"Error": "application/json response only"
}
```

Patch Boat

Update some attributes of a particular boat. Must be authorized by the boat user’s jwt.

PATCH /boats/<boat_id>

Request

Path Parameters (optional)

Name	Required	Description
boat_id	Yes	<string> String representation of an <integer> id of a boat.

Request Headers (required)

```
Accept: application/json

Authorization: <jwt>
```

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Required	Description
name	No*	<string> The name of the boat
type	No*	<string> The type of the boat.
length	No*	<integer> The length of the boat.

*Must have at least one attribute

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	Request body was missing required attributes.
Failure	401 Unauthorized	Request header was missing authorization jwt.
Failure	403 Forbidden	Request jwt was validated, but not the correct user associated with the requested boat. Boat is owned by someone else.
Failure	404 Not Found	Boat does not exist.
Failure	405 Method Not Allowed	Request method not allowed by endpoint. Automatic response by python Flask, in text/html format.
Failure	406 Not Acceptable	Request did not have accept application/json header.

Response Examples

Success. Status: 200 OK

```
{
"name": "Sea Witch",
"type": "Catamaran",
"length": 28,
"slip": null,
"owner": "auth0|655c254031108fd1ba1599af",
"id": 5190718477828096,
"self": "https://clean-utility-402919.uw.r.appspot.com/boats/5190718477828096"
}
```

Failure. Status: 400 Bad Request

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

Failure. Status: 401 Unauthorized

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

Failure. Status: 403 Forbidden

```
{
"Error": "Boat is owned by someone else."
}
```

Failure. Status: 404 Not Found

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

Failure. Status: 405 Method Not Allowed

```
<!doctype html>
<html lang=en>
<title>405 Method Not Allowed</title>
<h1>Method Not Allowed</h1>
<p>The method is not allowed for the requested URL.</p>
```

Failure. Status 406 Not Acceptable

```
{
"Error": "application/json response only"
}
```

Post Slips

Create a slip.

POST /slips

Request

Path Parameters

None

Request Headers (required)

```
Accept: application/json
```

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Required	Description
harbor	Yes	<string> The name of the harbor hosting the slip.
depth	Yes	<integer> The depth of water at the slip at mean low low water tide.
rate	Yes	<integer> The fee charge rate to dock a boat at this slip.

Request Body Example

```
{
"harbor": "Bar Harbor",
"depth": 50,
"rate": 100
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	**Status Code**	**Notes**
Success	201 Created	
Failure	400 Bad Request	Request body was missing required attributes.
Failure	405 Method Not Allowed	Request method not allowed by endpoint. Automatic response by python Flask, in text/html format.
Failure	406 Not Acceptable	Request did not have accept application/json header.

Response Examples

Success. Status: 201 Created

```
{
"harbor": "Bar Harbor",
"depth": 50,
"rate": 100,
"boat": null,
"id": 5112779182702592,
"self": "https://clean-utility-402919.uw.r.appspot.com/slips/5112779182702592"
}
```

Failure. Status: 400 Bad Request

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

Failure. Status: 405 Method Not Allowed

```
<!doctype html>
<html lang=en>
<title>405 Method Not Allowed</title>
<h1>Method Not Allowed</h1>
<p>The method is not allowed for the requested URL.</p>
```

Failure. Status 406 Not Acceptable

```
{
"Error": "application/json response only"
}
```

Get Slips

Gets all the slips. Authorization is not required to view slips occupied by boats. Non authorized users can see what boat is in an occupied slip, but they can’t view any other details about the boat.

GET /slips?offset=<offset>&limit=<limit>

Request

Path Parameters

Name	Required	Description
offset	No	<string> String representation of an <integer> for pagination offset. Default value = 0.
limit	No	<string> String representation of an <integer> for pagination limit. Default value = 5.

Request Headers (required)

```
Accept: application/json
```

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	405 Method Not Allowed	Request method not allowed by endpoint. Automatic response by python Flask, in text/html format.
Failure	406 Not Acceptable	Request did not have accept application/json header.

Response Examples

Success. Status: 200 OK

```
{
"slips": [
{
"boat": null,
"rate": 100,
"depth": 50,
"harbor": "Bar Harbor",
"id": 5112779182702592,
"self": "https://clean-utility-402919.uw.r.appspot.com/slips/5112779182702592"
}
],
"next": null
}
```

Failure. Status: 405 Method Not Allowed

```
<!doctype html>
<html lang=en>
<title>405 Method Not Allowed</title>
<h1>Method Not Allowed</h1>
<p>The method is not allowed for the requested URL.</p>
```

Failure. Status 406 Not Acceptable

```
{
"Error": "application/json response only"
}
```

Get Slip

Get a particular slip.

GET /slips/<slip_id>

Request

Path Parameters (optional)

Name	Required	Description
slip_id	Yes	<string> String representation of an <integer> id of a slip.

Request Headers (required)

```
Accept: application/json
```

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	Slip does not exist.
Failure	405 Method Not Allowed	Request method not allowed by endpoint. Automatic response by python Flask, in text/html format.
Failure	406 Not Acceptable	Request did not have accept application/json header.

Response Examples

Success. Status: 200 OK

```
{
"boat": null,
"rate": 100,
"depth": 50,
"harbor": "Bar Harbor",
"id": 5112779182702592,
"self": "https://clean-utility-402919.uw.r.appspot.com/slips/5112779182702592"
}
```

Failure. Status: 404 Not Found

```
{
"Error": "No slip with this slip_id exists"
}
```

Failure. Status: 405 Method Not Allowed

```
<!doctype html>
<html lang=en>
<title>405 Method Not Allowed</title>
<h1>Method Not Allowed</h1>
<p>The method is not allowed for the requested URL.</p>
```

Failure. Status 406 Not Acceptable

```
{
"Error": "application/json response only"
}
```

Delete Slip

Delete a particular slip. If the slip is occupied by a boat, then must be authorized by the boat user’s jwt.

DELETE /slips/<slip_id>

Request

Path Parameters (optional)

Name	Required	Description
slip_id	Yes	<string> String representation of an <integer> id of a slip.

Request Headers (required)

```
Accept: application/json

Authorization: <jwt> (required if occupied by boat)
```

Request Body

None

Response

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	When trying to delete a slip occupied by boat: request header was missing authorization jwt.
Failure	403 Forbidden	When trying to delete a slip occupied by boat: request jwt was validated, but not the correct user associated with the requested boat occupying slip. Boat is owned by someone else.
Failure	404 Not Found	Slip does not exist.
Failure	405 Method Not Allowed	Request method not allowed by endpoint. Automatic response by python Flask, in text/html format.

Response Examples

Success. Status: 204 No Content

Failure. Status: 401 Unauthorized

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

Failure. Status: 403 Forbidden

```
{
"Error": "Slip has boat owned by someone else."
}
```

Failure. Status: 404 Not Found

```
{
"Error": "No slip with this slip_id exists"
}
```

Failure. Status: 405 Method Not Allowed

```
<!doctype html>
<html lang=en>
<title>405 Method Not Allowed</title>
<h1>Method Not Allowed</h1>
<p>The method is not allowed for the requested URL.</p>
```

Put Slip

Update all attributes of a particular slip. If the slip is occupied by a boat, then must be authorized by the boat user’s jwt.

PUT /slips/<slip_id>

Request

Path Parameters (optional)

Name	Required	Description
slip_id	Yes	<string> String representation of an <integer> id of a slip.

Request Headers (required)

Accept: application/json

Authorization: <jwt> (required if slip is occupied by a boat)

Request Body

Required

Request Body Format

JSON

Request JSON Attribute

Name	Required	Description
harbor	Yes	<string> The name of the harbor hosting the slip.
depth	Yes	<integer> The depth of water at the slip at mean low low water tide.
rate	Yes	<integer> The fee charge rate to dock a boat at this slip.

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	Request body was missing required attributes.
Failure	401 Unauthorized	Request header was missing authorization jwt.
Failure	403 Forbidden	Request jwt was validated, but not the correct user associated with the requested boat occupying slip. Boat is owned by someone else.
Failure	404 Not Found	Slip does not exist.
Failure	405 Method Not Allowed	Request method not allowed by endpoint. Automatic response by python Flask, in text/html format.
Failure	406 Not Acceptable	Request did not have accept application/json header.

Response Examples

Success. Status: 200 OK

```
{
"boat": {
"id": 4921336719736832,
"self": "https://clean-utility-402919.uw.r.appspot.com/boats/4921336719736832"
},
"rate": 100,
"depth": 50,
"harbor": "Bar Harbor",
"id": 6570688919896064,
"self": "https://clean-utility-402919.uw.r.appspot.com/slips/6570688919896064"
}
```

Failure. Status: 400 Bad Request

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

Failure. Status: 401 Unauthorized

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

Failure. Status: 403 Forbidden

```
{
"Error": "Slip has boat owned by someone else."
}
```

Failure. Status: 404 Not Found

```
{
"Error": "No slip with this slip_id exists"
}
```

Failure. Status: 405 Method Not Allowed

```
<!doctype html>
<html lang=en>
<title>405 Method Not Allowed</title>
<h1>Method Not Allowed</h1>
<p>The method is not allowed for the requested URL.</p>
```

Failure. Status 406 Not Acceptable

```
{
"Error": "application/json response only"
}
```

Patch Slip

Update some attributes of a particular slip. If the slip is occupied by a boat, then must be authorized by the boat user’s jwt.

PATCH /slips/<slip_id>

Request

Path Parameters (optional)

Name	**Required**	**Description**
slip_id	Yes	<string> String representation of an <integer> id of a slip.

Request Headers (required)

Accept: application/json

Authorization: <jwt> (required if slip is occupied by a boat)

Request Body

Required

Request Body Format

JSON

Request JSON Attribute

Name	**Required**	**Description**
harbor	No*	<string> The name of the harbor hosting the slip.
depth	No*	<integer> The depth of water at the slip at mean low low water tide.
rate	No*	<integer> The fee charge rate to dock a boat at this slip.

*At least one attribute is required.

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	Request body was missing required attributes.
Failure	401 Unauthorized	Request header was missing authorization jwt.
Failure	403 Forbidden	Request jwt was validated, but not the correct user associated with the requested boat occupying slip. Boat is owned by someone else.
Failure	404 Not Found	Slip does not exist.
Failure	405 Method Not Allowed	Request method not allowed by endpoint. Automatic response by python Flask, in text/html format.
Failure	406 Not Acceptable	Request did not have accept application/json header.

Response Examples

Success. Status: 200 OK

```
{
"boat": {
"id": 4921336719736832,
"self": "https://clean-utility-402919.uw.r.appspot.com/boats/4921336719736832"
},
"rate": 100,
"depth": 50,
"harbor": "Bar Harbor",
"id": 6570688919896064,
"self": "https://clean-utility-402919.uw.r.appspot.com/slips/6570688919896064"
}
```

Failure. Status: 400 Bad Request

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

Failure. Status: 401 Unauthorized

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

Failure. Status: 403 Forbidden

```
{
"Error": "Slip has boat owned by someone else."
}
```

Failure. Status: 404 Not Found

```
{
"Error": "No slip with this slip_id exists"
}
```

Failure. Status: 405 Method Not Allowed

```
<!doctype html>
<html lang=en>
<title>405 Method Not Allowed</title>
<h1>Method Not Allowed</h1>
<p>The method is not allowed for the requested URL.</p>
```

Failure. Status 406 Not Acceptable

```
{
"Error": "application/json response only"
}
```

Put Boat into Slip

Put a boat in a particular slip. Must be authorized by the boat user’s jwt. Also removes the boat from any slip it was occupying before.

PUT /slips/<slip_id>/boats/<boat_id>

Request

Path Parameters (optional)

Name	Required	Description
slip_id	Yes	<string> String representation of an <integer> id of a slip.
boat_id	Yes	<string> String representation of an <integer> id of a boat.

Request Headers (required)

```
Accept: application/json

Authorization: <jwt>
```

Request Body

None

Response

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	Request header was missing authorization jwt.
Failure	403 Forbidden	Request jwt was validated, but not the correct user associated with the requested boat. Boat is owned by someone else.
Failure	403 Forbidden	Slip is already occupied by a boat.
Failure	404 Not Found	Either the Boat or the Slip does not exist.
Failure	405 Method Not Allowed	Request method not allowed by endpoint. Automatic response by python Flask, in text/html format.

Response Examples

Success. Status: 204 No Content

Failure. Status: 401 Unauthorized

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

Failure. Status: 403 Forbidden

```
{
"Error": "Boat is owned by someone else."
}
```

Failure. Status: 403 Forbidden

```
{
"Error": "The slip is not empty or boat is already at slip"
}
```

Failure. Status: 404 Not Found

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

Failure. Status: 404 Not Found

```
{
"Error": "No slip with this slip_id exists"
}
```

Failure. Status: 405 Method Not Allowed

```
<!doctype html>
<html lang=en>
<title>405 Method Not Allowed</title>
<h1>Method Not Allowed</h1>
<p>The method is not allowed for the requested URL.</p>
```

Remove Boat into Slip

Remove a boat from a slip. Must be authorized by the boat user’s jwt.

DELETE /slips/<slip_id>/boats/<boat_id>

Request

Path Parameters (optional)

Name	Required	Description
slip_id	Yes	<string> String representation of an <integer> id of a slip.
boat_id	Yes	<string> String representation of an <integer> id of a boat.

Request Headers (required)

```
Accept: application/json

Authorization: <jwt>
```

Request Body

None

Response

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	401 Unauthorized	Request header was missing authorization jwt.
Failure	403 Forbidden	Request jwt was validated, but not the correct user associated with the requested boat. Boat is owned by someone else.
Failure	404 Not Found	Either the Boat or the Slip does not exist, or they both exist but the Boat requested is not at the Slip.
Failure	405 Method Not Allowed	Request method not allowed by endpoint. Automatic response by python Flask, in text/html format.

Response Examples

Success. Status: 204 No Content

Failure. Status: 401 Unauthorized

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

Failure. Status: 403 Forbidden

```
{
"Error": "Boat is owned by someone else."
}
```

Failure. Status: 404 Not Found

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

Failure. Status: 404 Not Found

```
{
"Error": "No slip with this slip_id exists"
}
```

Failure. Status: 404 Not Found

```
{
"Error": "No boat with this boat_id is at the slip with this slip_id"
}
```

Failure. Status: 405 Method Not Allowed

```
<!doctype html>
<html lang=en>
<title>405 Method Not Allowed</title>
<h1>Method Not Allowed</h1>
<p>The method is not allowed for the requested URL.</p>
```