

Portfolio Assignment

Deployed: <https://cs493final-rovirose.wm.r.appspot.com/>
Log In: <https://cs493final-rovirose.wm.r.appspot.com/welcome>

Emmanuel Rovirosa
CS 493: Cloud Application Development
Spring 2020
Oregon State University

View all Users

List all the boats for the authenticated user.

GET /users

Request

Request 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
[
  {
    "Email": "emmanuel.rovirosa@gmail.com",
    "id": "110116583805307179072"
  },
  {
    "Email": "rovirose@oregonstate.edu",
    "id": "110206237738116176759"
  }
]
```

Create a Boat

Allows you to create a new boat.

POST /boats

Request

Request Parameters

None

Request Body

Required

Request Header

Authorization – Bearer Token (required)

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
name	String	The name of the boat.	Yes
type	String	The type of the boat. E.g., Sailboat, Catamaran, etc.	Yes
length	Integer	Length of the boat in feet.	Yes

Request Body Example

<pre>{ "name": "Sea Witch", "type": "Catamaran", "length": 28 }</pre>

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 other than the ones that are listed).</p>
Failure	401 Unauthorized	The JWT was missing or expired

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 `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. You must not store the `self` attribute in Datastore

Success

Status: 201 Created

```
{
  "id": "abc123",
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "self": "https://<your-app>/boats/abc123"
}
```

Failure

Status: 400 Bad Request

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

Status: 401 Bad Request

```
{
  "Error": "Missing or invalid JWT"
}
```

Get a Boat

Allows you to get an existing boat if it is sent with the owner's jwt

GET /boats/:boat_id

Request

Request Parameters

Name	Type	Description	Required?
boat_id	String	ID of the boat	Yes

Request Body

None

Request Header

Authorization – Bearer Token (required)

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	The JWT was missing or expired
Failure	403 Forbidden	Request not made by owner of the boat
Failure	404 Not Found	No boat with this boat_id exists

Response Examples

Success

Status: 200 OK
<pre>{ "id": "abc123", # automatically generated by Datastore "name": "Sea Witch", #The name of the boat, a string "type": "Catamaran", #The type of boat, power boat, sailboat, catamaran etc. a string "length": 28, #The length of the boat "slips": [{ "id": "123abc", "self": "https://appspot.com/slips/123abc" }, { "id": "456", "self": "https://appspot.com/slips/456" }], # This embedded list does NOT need to paginate. "self": "https://appspot.com/boats/abc123" }</pre>

Failure

Status: 401 Unauthorized

```
{  
"Error": "Missing or invalid JWT"  
}
```

Status: 403 Forbidden

```
{  
"Error": "The user with this jwt does not own a boat with this boat ID"  
}
```

Status: 404 Not Found

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

View all Boats

List all the boats for the authenticated user.

GET /boats

Request

Request Parameters

None

Request Header

Authorization – Bearer Token (required)

Query Parameters

Name	Type	Description	Required?
limit	integer	Max page size	No
offset	integer	Index of first item of the page	No

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	401 Unauthorized	The JWT was missing or expired

Response Examples

Success

Status: 200 OK
<pre>{ "boats": [{ "slips": [{ "self": "http://localhost:5000/slips/5423713700806656", "id": 5423713700806656 }], "length": 200, "type": "Mega Yatch", "name": "Odyssey II", "id": 4688579959717888 }, { "length": 99, "type": "Yatch",</pre>

```
    "name": "Odyssey",
    "id": 4813393923932160
  },
  {
    "name": "Odyssey",
    "length": 99,
    "type": "Yatch",
    "id": 5173791541952512
  }
],
"next": "http://localhost:5000/boats?limit=3&offset=3"
}
```

Note: next field will only appear if there are more pages

Failure

Status: 401 Bad Request

```
{
  "Error": "Missing or invalid JWT"
}
```

Edit a Boat

Allows you to edit a boat if it belongs to the authenticated user

PATCH /boats/:boat_id

Request

Request Parameters

Name	Type	Description	Required?
boat_id	String	ID of the boat	Yes

Request Header

Authorization – Bearer Token (required)

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
name	String	The name of the boat.	Yes
type	String	The type of the boat. E.g., Sailboat, Catamaran, etc.	Yes
length	Integer	Length of the boat in feet.	Yes

Request Body Example

```
{
  "name": "Sea Witch",
  "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.
Failure	401 Unauthorized	The JWT was missing or expired
Failure	403 Forbidden	Request not made by owner of the boat

Response Examples

- The `self` attribute will contain the live link to the REST resource corresponding to this boat. You must not store this attribute in Datastore.

Success

Status: 200 OK


```
{
  "id": "abc123",
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 99,
  "self": "https://<your-app>/boats/abc123"
}
```

Failure

Status: 400 Bad Request

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

Status: 401 Bad Request

```
{
  "Error": "Missing or invalid JWT"
}
```

Status: 403 Forbidden

```
{
  "Error": "The user with this jwt does not own a boat with this boat ID"
}
```

Delete a Boat

Allows you to delete a boat.

DELETE /boats/:boat_id

Request

Request Parameters

None

Request Header

Authorization – Bearer Token (required)

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	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.
Failure	401 Unauthorized	The JWT was missing or expired
Failure	403 Forbidden	Request not made by owner of the boat

Response Examples

Success

Status: 204 No Content

Failure

<p>Status: 400 Bad Request</p> <pre>{ "Error": "The request object is missing at least one of the required attributes" }</pre> <p>Status: 401 Bad Request</p> <pre>{ "Error": "Missing or invalid JWT" }</pre> <p>Status: 403 Forbidden</p> <pre>{</pre>
--

```
"Error": " The user with this jwt does not own a boat with this boat ID"
}
```

Create a Slip

Allows you to create a new slip.

POST /slips

Request

Request Parameters

None

Request Body

Required

Request Body Format

JSON

Request JSON Attributes

Name	Type	Description	Required?
number	Integer	Number of the slip	Yes
length	Integer	Maximum length of boat for that slip	Yes
width	Integer	Maximum width of boat for that slip	Yes

Request Body Example

```
{
  "number": 1,
  "length": 20,
  "width": 5
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	201 Created	
Failure	400 Bad Request	If the request is missing a required attribute, and 400 status code must be returned.

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 "current_boat" is the ID of the boat currently at this slip. If there is no boat in this slip, the value of "current_boat" should be null
- The value of the attribute self is a live link to the REST resource corresponding to this slip. In other words, this is the URL to get this newly created slip. Success

```
Status: 201 Created
{
  "number": 1,
```

```
"length": 20,  
"width": 5,  
"id": 5186940114567168,  
"current_boat": null,  
"self": "http://127.0.0.1:8080/slips/5186940114567168"  
}
```

Failure

Status: 400 Bad Request

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

View a slip

Allows you to get an existing slip.

GET /slips/:slip_id

Request

Request Parameters

Name	Type	Description	Required?
slip_id	String	ID of the slip	Yes

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No slip with this slip_id exists

Response Examples

Success

Status: 200 OK
<pre>{ "width": 5, "number": 1, "length": 20, "id": 5186940114567168, "self": "http://127.0.0.1:8080/slips/5186940114567168", "current_boat": null }</pre>

Failure

Status: 404 Not Found
<pre>{ "Error": "No slip with this slip_id exists" }</pre>

List all slips

List all the slips.

GET /slips

Request

Request Parameters

None

Query Parameters

Name	Type	Description	Required?
limit	integer	Max page size	No
offset	integer	Index of first item of the page	No

Request Body

None

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	

Response Examples

Success

Status: 200 OK
<pre>[{ "width": 5, "number": 1, "length": 20, "id": 5107756889538560, "current_boat": null }, { "number": 1, "length": 20, "width": 5, "id": 5186940114567168, "current_boat": null }, { "length": 20, "current_boat": 5764518391054336, "width": 5, "number": 1,</pre>

```
    "id": 5655374346584064
  },
  {
    "number": 1,
    "length": 20,
    "width": 5,
    "current_boat": 5092424393162752,
    "id": 5680529164730368
  }
]
```


Delete a slip

Allows you to delete a slip. If the Slip being deleted contained a boat then, the boat will be updated.

DELETE /Slips/:Slip_id

Request

Request Parameters

None

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 slip with this slip_id exists

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found <pre>{ "Error": "No Slip with this Slip_id exists" }</pre>

Assign a Boat to a Slip

Boat has been assigned to dock in a slip

```
PUT /slip/:slip_id:boat_id
```

Request

Request Parameters

None

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 slip exists with this slip_id and the slip is not being carried
Failure	403 Forbidden	Slip is not empty
Failure	404 Not Found	No boat with this boat_id exists, and/or no slip with this slip_id exists.

Response Examples

Success

```
Status: 204 No Content
```

Failure

```
Status: 403 Forbidden

{
  "Error": "The slip is not empty"
}
```

```
Status: 404 Not Found

{
  "Error": "The specified boat and/or slip don't exist"
}
```

Remove a Boat from a Slip

Removes a boat from a slip. This will current_boat the carrier from the slip and the current_slip from the boat.

DELETE /slip/:slip_id/boat_id

Request

Request Parameters

None

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 slip exists with this slip_id and the boat is carrying the slip
Failure	404 Not Found	No boat with this boat_id is at the slip with this slip_id.

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found
<pre>{ "Error": "No boat with this boat_id is at the slip with this slip_id" }</pre>