Assignment 5: API Spec

CS 493: Cloud Application Development

Clint Hawkes

Oregon State University

Change log

Page	Change	Date
	Initial version.	M, Oct 12

App URL: https://hw5-hawkesc.wl.r.appspot.com

Error Message for All URLs

If a method is not allowed for the target URL:

Response Body Format

JSON

```
Status: 405 Method Not Allowed

{
"Error":  "Method not allowed"
}
```

Create a Boat

Allows you to create a new boat.

```
POST /boats
```

Request

Path Parameters

None

Request Body

Required

Request Body Format

JSON - ***If JSON is incorrectly formatted and cannot be parsed, server will return generic error

Request JSON Attributes

Name	Type	Description	Required?
name	String	The name of the boat. Must NOT already be in use. String must contain 50 or fewer characters. String may consist of letters, numbers, spaces, and the following characters - _ ? ! . , $ # @ & / % *	Yes
type	String	The type of the boat. E.g., Sailboat, Catamaran, etc. String must contain 50 or fewer characters. String must only consist of letters, numbers and spaces.	Yes
length	Integer	Length of the boat in feet. Must not contain decimal numbers. Value must be greater than 0 and less than 2000.	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	If the request is missing any of the 3 required attributes, the boat must not be updated, and 400 status code must be returned. All attributes in the request body must adhere to the requirements outlined above in the "Request JSON Attributes" section. A 400 error message will be returned if they don't. Any extraneous attributes in the request body that are not "name", "type" or "length" will be ignored by the server.
Failure	406 Not Acceptable	Response must be in JSON. Client must be able to accept json format
Failure	415 Unsupported Media Format	Request object must be 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 `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

Missing Attribute:

```
Status: 400 Bad Request
```

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

Incorrect Attribute Format:

```
Status: 400 Bad Request

{
"Error": "One or more of the request attributes do not conform to the
required type/format"
}
```

Creating boat with duplicate name:

```
Status: 400 Bad Request

{
"Error": "That name is already in use"
}
```

Client Requests Unsupported Format

```
Status: 406 Not Acceptable

{
"Error": "The response must be json"
}
```

Client Request Wrong Type

```
Status: 415 Unsupported Media Type

{
"Error": "The request object must be json"
}
```

Get a Boat

Allows you to get an existing boat

GET /boats/:boat_id

Request

Path Parameters

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

Request Body

None

Response

Response Body Format

JSON or HTML

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	Boat ID cannot be null
Failure	404 Not Found	No boat with this boat_id exists
Failure	406 Not Acceptable	Response must be in JSON or HTML. Client must be able to accept one of these two formats.

Response Examples

Success

Response in JSON

```
Status: 200 OK
{
  "id": "abc123",
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 28,
  "self": "https://<your-app>/boats/abc123"
}
```

Response in HTML

```
Status: 200 OK
<table border="1">
<tr>
<th>length</th>
<td>28</td>
</tr>
<tr>
<th>name</th>
<td>Sea Witch</td>
</tr>
<tr>
<th>type</th>
<td>Catamaran</td>
</tr>
<tr>
<th>id</th>
<td>abc123</td>
</tr>
<tr>
<th>self</th>
<td>http://<your-app>/boats/abc123</td>
</tr>
</table>
```

Failure

Null Boat ID:

```
Status: 400 Bad Request

{
"Error":  "Boat ID must not be null"
}
```

Invalid Boat ID

```
Status: 404 Not Found
{
"Error":  "No boat with this boat_id exists"
}
```

Client Requests Unsupported Format

```
Status: 406 Not Acceptable

{
"Error":  "The response must be json or html"
}
```

Edit a Boat - Subset of attributes

Allows you to edit a boat.

PATCH /boats/:boat_id

Request

Path Parameters

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

Request Body

Required

Request Body Format

JSON - ***If JSON is incorrectly formatted and cannot be parsed, server will return generic error

Request JSON Attributes

Name	Type	Description	Required?
name	String	The name of the boat. Must NOT already be in use. String must contain 50 or fewer characters. String may consist of letters, numbers, spaces, and the following characters - _ ? ! . , $ # @ & / % *	No
type	String	The type of the boat. E.g., Sailboat, Catamaran, etc. String must contain 50 or fewer characters. String must only consist of letters, numbers and spaces.	No
length	Integer	Length of the boat in feet. Must not contain decimal numbers. Value must be greater than 0 and less than 2000.	No

***Attributes not included in the request will remain unchanged.

Request Body Example

```
{
  "name": "Sea Witch",
  "type": "Catamaran"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	400 Bad Request	All attributes in the request body must adhere to the requirements outlined above in the "Request JSON Attributes" section. An error message will be returned if they don't. Any extraneous attributes in the request body that are not in the boat entity will be ignored by the server. Boat ID in the URL path must not be null.
Failure	403 Forbidden	Not allowed to change the boat ID
Failure	404 Not Found	No boat with this boat_id exists
Failure	406 Not Acceptable	Response must be in JSON
Failure	415 Unsupported Media Type	Request must be in JSON

Response Examples

Success

```
Status: 200 OK

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

Failure

Incorrect Name Attribute Format:

```
Status: 400 Bad Request

{
```

```
"Error": "The 'name' attribute in the request not conform to the required
type/format"
}
```

Incorrect Type Attribute Format:

```
Status: 400 Bad Request

{
"Error": "The 'type' attribute in the request not conform to the required
type/format"
}
```

Incorrect Length Attribute Format:

```
Status: 400 Bad Request

{
"Error": "The 'length' attribute in the request not conform to the required
type/format"
}
```

Null Boat ID:

```
Status: 400 Bad Request

{
"Error": "Boat ID must not be null"
}
```

Changing name to one in use:

```
Status: 400 Bad Request

{
"Error": "That name is already in use"
}
```

Changing the Boat ID

```
Status: 403 Forbidden

{
"Error": "You are not allowed to change the boat ID"
}
```

Incorrect Boat ID

```
Status: 404 Not Found

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

Client Requests Unsupported Format

```
Status: 406 Not Acceptable

{
"Error":  "The response must be json"
}
```

Client Request Wrong Type

```
Status: 415 Unsupported Media Type

{
"Error":  "The request object must be json"
}
```

Edit a Boat - All attributes

Allows you to edit a boat.

PUT /boats/:boat_id

Request

Path Parameters

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

Request Body

Required

Request Body Format

JSON - ***If JSON is incorrectly formatted and cannot be parsed, server will return generic error

Request JSON Attributes

Name	Type	Description	Required?
name	String	The name of the boat. Must **NOT** already be in use. String must contain 50 or fewer characters. String may consist of letters, numbers, spaces, and the following characters - _ ? ! . , $ # @ & / % *	Yes
type	String	The type of the boat. E.g., Sailboat, Catamaran, etc. String must contain 50 or fewer characters. String must only consist of letters, numbers and spaces.	Yes
length	Integer	Length of the boat in feet. Must not contain decimal numbers. Value must be greater than 0 and less than 2000.	Yes

Request Body Example

```
{
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 99
}
```

Response

Response Body Format

Success: No body

Failure: JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 okay	
Success	303 See Other	If the browser does not redirect automatically, the response body will be empty. The URL to the updated resource can be found in the Location header of the response.
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. All attributes in the request body must adhere to the requirements outlined above in the "Request JSON Attributes" section. An error message will be returned if they don't. Any extraneous attributes in the request body that are not found in the boat entity will be ignored by the server. Boat ID in the URL path must not be null.
Failure	403 Forbidden	Not allowed to change the boat ID
Failure	404 Not Found	No boat with this boat_id exists
Failure	415 Unsupported Media Type	Request body attributes must be application/json format.

Response Examples

Success - Redirected

```
Status: 200 OK
{
  "id": "abc123",
  "name": "Sea Witch",
  "type": "Catamaran",
  "length": 99,
  "self": "https://<your-app>/boats/abc123"
}
```

Success – No Redirection

Status: 303 See Other

***This response is due to a redirection after a 303 code is returned. URL to updated boat is found under the Location header of the response, and browser/tool should automatically redirect and display the boat information as shown above.

Failure

Missing Attribute:

```
Status: 400 Bad Request
{
"Error": "The request object is missing at least one of the required
attributes"
}
```

Incorrect Attribute Format:

```
Status: 400 Bad Request
{
"Error": "One or more of the request attributes do not conform to the
required type/format"
}
```

Changing name to one in use:

```
Status: 400 Bad Request

{
"Error": "That name is already in use"
}
```

Null Boat ID:

```
Status: 400 Bad Request

{
"Error": "Boat ID must not be null"
}
```

Changing the Boat ID

```
Status: 403 Forbidden

{
"Error": "You are not allowed to change the boat ID"
}
```

Incorrect Boat ID

```
Status: 404 Not Found


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

Client Request Wrong Type

```
Status: 415 Unsupported Media Type


{
"Error":  "The request object must be json"
}
```

Delete a Boat

Allows you to delete a boat. Note that if the boat is currently in a slip, deleting the boat makes the slip empty.

DELETE /boats/:boat_id

Request

Path Parameters

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

Request Body

None

Response

No body

Response Body Format

Success: No body

Failure: JSON error message

Response Statuses

Outcome	Status Code	Notes
Success	204 No Content	
Failure	400 Bad Request	Boat ID can not be null
Failure	404 Not Found	No boat with this boat_id exists

Response Examples

Success

Status: 204 No Content

Failure

Null Boat ID:

Status: 400 Bad Request
{

```
"Error": "Boat ID must not be null"
}
```

```
Status: 404 Not Found

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