

Assignment 5: API Spec

URL:

<https://arifig-osu-hw5.ue.r.appspot.com/>

CS 493: Cloud Application Development

Spring 2020

Oregon State University

Last Update: May 1, 2020. 9:00 am Pacific

Change log.....	1
Create a Boat.....	2
Get a Boat.....	3
Patch a boat.....	4
Put a boat.....	5
List all boats.....	8
Delete a boat.....	9

Change log

Page	Change	Date
	Initial version.	Sn, May 1

Create a Boat

Allows you to create a new boat.

POST /boats

Request

Request Parameters

None

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
loads	array	Length of the boat in feet.	No

Request Body Example

```
{
  "name": "g1",
  "type": "test",
  "length": 99
}
```

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	415 Unsupported Media Type	When the clients request body is not in the JSON format
Failure	403 Forbidden	If a boat with that name already exists
Failure	400 Bad Request	If a boat name isn't a string, or the boat name is greater than 30 characters. Also, if a boat type isn't a string or is greater than 30 characters. Finally, if a boat length is not an integer or is greater than 9999.

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": 5643280054222848,
  "length": 99,
  "name": "g2",
  "self": "http://127.0.0.1:8080/boats/5643280054222848",
  "type": "test"
}
```

Failure

Status: 400 Bad Request

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

Failure

Status: 415 Unsupported Media Type

```
{
  "Error": "Incorrect media type, only JSON format accepted"
}
```

Failure

Status: 403 Forbidden

```
{
  "Error": "A boat with that name already exists"
}
```

Failure

```
Status: 400 Bad Request
(Validation checks)
{
  'Error': 'Boat type must be a string'
}
or
{'Error': 'Type must be 30 characters or less'}
or
{'Error': 'Boat length must be an integer'}
or
{'Error': 'Boat length must be smaller than 9999'}
or
{'Error': 'Boat name must be a string'}
or
{'Error': 'Boat name must be 30 characters or less'}
```

Notes

- Only requests with a format of JSON are accepted.
- Boat names and types must be 30 characters or less
- Boat lengths must be less than 9999
- Boat lengths must be integers.
- Boat names and types must be strings

Get a Boat

Allows you to get an existing boat

```
GET /boats/:boat_id
```

Request

Request Parameters

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

Request Body

None

Response

Response Body Format

JSON, text/html

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No boat with this boat_id exists

Failure	406 Not Acceptable	The media type requested in the accept header is not supported.
---------	--------------------	---

Response Examples

Success

Status: 200 OK JSON

```
{
  "id": 5643280054222848,
  "length": 99,
  "name": "g2",
  "self": "http://127.0.0.1:8080/boats/5643280054222848",
  "type": "test"
}
```

Success

Status: 200 OK text/html

```
<table border="1">
  <tr>
    <th>name</th>
    <td>g2</td>
  </tr>
  <tr>
    <th>length</th>
    <td>99</td>
  </tr>
  <tr>
    <th>type</th>
    <td>test</td>
  </tr>
  <tr>
    <th>self</th>
    <td>http://127.0.0.1:8080/boats/5643280054222848</td>
  </tr>
  <tr>
    <th>id</th>
    <td>5643280054222848</td>
  </tr>
</table>
```

Failure

Status: 404 Not Found

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

Failure

Status: 406 Not Allowable

```
{
  'Error': 'Media type requested not supported. Supported mime types are:
JSON, text/html'
}
```

PATCH a Boat

Allows you to edit any number of properties of an existing boat

GET /boats/:boat_id

Request

Request Parameters

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

Request Body

JSON

Request Body Example

```
{
  "type": "test-PATCH"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	200 OK	
Failure	404 Not Found	No boat with this boat_id exists
Failure	415 Not Acceptable	The media type in the request body is not supported.
Failure	403 Forbidden	If a boat with that name already exists
Failure	400 Bad Request	If a boat name isn't a string, or the boat name is greater than 30 characters. Also, if a boat type isn't a string or is greater than 30 characters. Finally, if a boat length is not an integer or is greater than 9999.

Response Examples

Success

Status: 200 OK

```
{
  "id": 5643280054222848,
  "length": 99,
  "name": "g2",
  "self": "http://127.0.0.1:8080/boats/5643280054222848",
  "type": "test-PATCH"
}
```

Failure

Status: 404 Not Found

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

Failure

Status: 406 Not Allowable

```
{
  'Error': 'Media type requested not supported. Supported mime types are:
JSON, text/html'
}
```

Failure

Status: 400 Bad Request

(Validation checks)

```
{
  'Error': 'Boat type must be a string'
}
or
{'Error': 'Type must be 30 characters or less'}
or
{'Error': 'Boat length must be an integer'}
or
{'Error': 'Boat length must be smaller than 9999'}
or
{'Error': 'Boat name must be a string'}
or
{'Error': 'Boat name must be 30 characters or less'}
```

Notes

- Only requests with a format of JSON are accepted.
- Boat names and types must be 30 characters or less
- Boat lengths must be less than 9999
- Boat lengths must be integers.
- Boat names and types must be strings

PUT a Boat

Allows you to update all of the properties of an existing boat

GET /boats/:boat_id

Request

Request Parameters

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

Request Body

JSON

Request Body Example

```
{
  "length": 100,
  "name": "g2-PUT",
  "type": "test-PUT"
}
```

Response

Response Body Format

JSON

Response Statuses

Outcome	Status Code	Notes
Success	303 See Other	The location of the put boat is in the "Location" value of the response header
Failure	404 Not Found	No boat with this boat_id exists
Failure	415 Not Acceptable	The media type in the request body is not supported.
Failure	403 Forbidden	If a boat with that name already exists
Failure	400 Bad Request	When the put body is missing one of the required parameters (name, type, length)
Failure	400 Bad Request	If a boat name isn't a string, or the boat name is greater than 30 characters. Also, if a boat type isn't a string or is greater than 30 characters. Finally, if a boat length is not an integer or is greater than 9999.

Response Examples

Success

Status: 303 See Other
In Headers:
Content-Type →application/json

Location →http://127.0.0.1:8080/boats/5643280054222848
Content-Length →0
Server →Werkzeug/1.0.1 Python/3.6.9
Date →Fri, 01 May 2020 19:09:27 GMT

Failure

Status: 404 Not Found

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

Failure

Status: 406 Not Allowable

```
{  
  'Error': 'Media type requested not supported. Supported mime types are:  
JSON, text/html'  
}
```

Failure

Status: 400 Bad Request

(Validation checks)

```
{  
  'Error': 'Boat type must be a string'  
}  
or  
{'Error': 'Type must be 30 characters or less'}  
or  
{'Error': 'Boat length must be an integer'}  
or  
{'Error': 'Boat length must be smaller than 9999'}  
or  
{'Error': 'Boat name must be a string'}  
or  
{'Error': 'Boat name must be 30 characters or less'}  
or  
{"Error": 'Incorrect number of attributes [need: name, type, and length]'}
```

Notes

- Only requests with a format of JSON are accepted.
- Boat names and types must be 30 characters or less
- Boat lengths must be less than 9999
- Boat lengths must be integers.
- Boat names and types must be strings
- Request body must have all 3 of the required boat parameters (name, type, length)
- The updated boat is under 'Location' in the headers of the response.

List all Boats

List all the boats.

GET /boats

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

```
{
  "boats": [
    {
      "id": 5635008819625984,
      "length": 2,
      "name": "g4UT",
      "self":
"http://127.0.0.1:8080/boats/5635008819625984",
      "type": "test-PUT"
    },
    {
      "id": 5643280054222848,
      "length": 100,
      "name": "g2-PUT",
      "self":
"http://127.0.0.1:8080/boats/5643280054222848",
      "type": "test-PUT"
    },
    {

```

```
        "id": 5710353417633792,  
        "length": 1,  
        "name": "g3",  
        "self":  
        "http://127.0.0.1:8080/boats/5710353417633792",  
        "type": "test"  
    },  
    ],  
    "next": "http://127.0.0.1:8080/boats?  
limit=3&offset=3"  
}
```

Delete a Boat

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

DELETE /boats/: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	
Failure	404 Not Found	No boat with this boat_id exists

Response Examples

Success

Status: 204 No Content

Failure

Status: 404 Not Found

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